ESL-TR-88-24

ENVIRONMENTAL INTERACTIONS OF HYDRAZINE FUELS IN SOIL/WATER SYSTEMS

J. STREET, C. JOHNSTON, L. OU, R. MANSELL
S. BLOOM

SOIL SCIENCE DEPARTMENT
UNIVERSITY OF FLORIDA
GAINESVILLE FL 32611

OCTOBER 1988

FINAL REPORT

MARCH 1985 - SEPTEMBER 1987

ENGINEERING & SERVICES LABORATORY
AIR FORCE ENGINEERING & SERVICES CENTER
TYNDALL AIR FORCE BASE, FLORIDA 32403

NOTICE

PLEASE DO NOT REQUEST COPIES OF THIS REPORT FROM HQ AFESC/RD (ENGINEERING AND SERVICES LABORATORY). ADDITIONAL COPIES MAY BE PURCHASED FROM:

NATIONAL TECHNICAL INFORMATION SERVICE
5285 PORT ROYAL ROAD
SPRINGFIELD, VIRGINIA 22161

FEDERAL GOVERNMENT AGENCIES AND THEIR CONTRACTORS REGISTERED WITH DEFENSE TECHNICAL INFORMATION CENTER SHOULD DIRECT REQUESTS FOR COPIES OF THIS REPORT TO:

DEFENSE TECHNICAL INFORMATION CENTER
CAMERON STATION
ALEXANDRIA, VIRGINIA 22314

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION: UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT: Approved for Public Release Distribution Unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S): ESL-TR-88-24
6a. NAME OF PERFORMING ORGANIZATION: Soil Science Department; 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION: Air Force Engineering and Services Center
6c. ADDRESS (City, State, and ZIP Code): University of Florida, Gainesville FL 32611	7b. ADDRESS (City, State, and ZIP Code): HQ AFESC/RDVS, Tyndall AFB FL 32403-6001
8a. NAME OF FUNDING/SPONSORING ORGANIZATION: AFESC; 8b. OFFICE SYMBOL (If applicable): RDV	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER: F08635-83-C-0136
8c. ADDRESS (City, State, and ZIP Code): HQ AFESC/RDVS, Tyndall AFB FL 32403-6001	10. SOURCE OF FUNDING NUMBERS: PROGRAM ELEMENT NO. 6.1; PROJECT NO. 1900; TASK NO. 2072; WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
Environmental Interactions of Hydrazine Fuels in Soil/Water Systems
UNCLASSIFIED

12. PERSONAL AUTHOR(S)
Jimmy Street, Clifford Johnston, Li Ou, Robert Mansell, and Steve Bloom

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Final	FROM Mar 85 TO Sep 87	October 1988	

16. SUPPLEMENTARY NOTATION
Availability of this report is specified on reverse of front cover.

17. COSATI CODES		
FIELD	GROUP	SUB-GROUP
07	04	
08	04	

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Hydrazine; Monomethylhydrazine; Interactions; Transport Modeling; Degradation; Soil Column Studies; Copper ions; Biodeterioration; Environmental impact; Water pollution; Soil pollution

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The studies of the interactions of hydrazine in aqueous and soil systems are presented in this technical report. The studies are divided into the following five areas: aqueous and soil suspension studies, surface interaction studies, biological interaction studies, soil column studies, and soil transport modeling. The objective of this work is to determine the fate of hydrazine fuel released into an aqueous or soil environment.

Aqueous degradation studies reveal that the extent of hydrazine degradation and the products formed are highly dependent upon several variables. Among these include the type of container used in the studies, the presence of certain metal ions, the ionic strength, the presence and type of pH buffer, the temperature, the presence of bacteria, and the amount of dissolved oxygen. In particular, aqueous hydrazine degradation is particularly rapid in quartz vessels with Cu^{+2} ions and oxygen present. Degradation also increases with increasing ionic strength, pH buffer concentration, temperature, and bacteria content. In the presence of Cu^{+2} ions, ammonia is formed as a product.

Keywords

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL: Capt Floyd L. Wiseman	22b. TELEPHONE (Include Area Code): (904) 283-4234; 22c. OFFICE SYMBOL: HQ AFESC/RDVS

DD Form 1473, JUN 86 — Previous editions are obsolete. — SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

In soil suspensions of montmorillonite and kaolinite clays, cationic exchange at pH 4 and 8 is a principle interaction of hydrazine. At pH 4, hydrazine exists mostly as hydrazinium ($N_2H_5^+$), whereas at pH 8, hydrazine exists in its neutral form (N_2H_4). At the high pH, more hydrazine adsorption and surface catalysis occurs. In suspensions of the top horizon of Arredondo soil, only 20 percent of the hydrazine is recovered at pH 8; 80 percent is recovered at pH 4.

Infrared and Raman spectroscopy were used to study the microscopic interactions of hydrazine with kaolinite. Studies show that, after 2 hours of hydrazine exposure to kaolinite, 90 percent of the spacing between clay layers is intercalated by hydrazine. X-ray diffraction shows the layer spacing increases from .716 to 1.03 nanometers. At reduced pressures, spectroscopic evidence shows that the hydrazine penetrates into the inner structure of the clay layers and interacts strongly with the inner hydroxy groups. In addition, the interlayer spacing decreases upon evacuation, giving further evidence that hydrazine penetrates deeper into the layers.

Microbiological studies show there are certain bacteria which can cometabolically degrade hydrazine. In general, however, biodegradation is a minor factor in the overall degradation of hydrazine in aqueous and soil systems. ^{15}N-hydrazine studies show that the Achromobacter species can degrade hydrazine to nitrogen. Soil suspensions treated with the Achromobacter species degrades hydrazine somewhat faster than suspensions without the bacteria. At high concentrations, hydrazine exerts a severe toxic effect on the bacteria, and biodegradation is therefore minimized.

Microbiological studies of monomethylhydrazine (MMH) show that the initial degradation steps of MMH are not enhanced by the presence of bacteria. In the absence of soil bacteria, MMH will degrade, but does not yield carbon dioxide (CO_2). However, ^{14}C-MMH studies show that MMH does degrade to yield CO_2 if soil bacteria is present.

Soil column studies show that the mobility of hydrazine through a water-saturated soil matrix depends upon several factors and that there are several types of interactions in soil. These interactions include ion exchange, adsorption and complexation, first- and second-order chemical reactions, and biodegradation. Mobility can depend upon such factors as the soil horizon, the initial input concentration, and the pore velocity. Studies were done at lower pH; therefore hydrazine existed as $N_2H_5^+$ in the soil columns. $N_2H_5^+$ is not as susceptible to catalytic oxidation as N_2H_4. Three Arredondo soil horizons were studied. The top horizon has the greatest organic and bacteria content; hence, the greatest interactions were found with this horizon.

Information from all the hydrazine interaction studies was used to develop a transport model to describe the behavior of hydrazine in the soil columns. The model includes diffusion and convection terms in conjunction with chemical adsorption and degradation terms. The model has two adsorption processes in series, and an irreversible process. The model was used to fit data from the soil column studies.

A more elaborate model describing the transport of hydrazine from a leak in a storage tank downstream to a drainage site is also presented in the report.

PREFACE

This report was prepared by the Soil Science Department at the University of Florida, Gainesville FL 32611. This work was sponsored by HQ Air Force Engineering and Services Center and Services Laboratory, Tyndall AFB FL 32403-6001. Capt Floyd L. Wiseman (AFESC/RDVS) was the government project officer. This report summarizes work accomplished between March 1985 and September 1987 under program element 62206F.

This report has been reviewed by the Public Affairs Office (PA) and is releasable to the National Technical Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nationals.

This technical report has been reviewed and is approved for publication:

FLOYD L. WISEMAN, Capt, USAF
Project Officer

THOMAS J. WALKER, Lt Col, USAF, BSC
Chief, Environics Division

KENNETH T. DENBLEYKER, Maj, USAF
Chief, Environmental Sciences Branch

LAWRENCE D. HOKANSON, Colonel, USAF
Director, Engineering and Services Laboratory

(The reverse of this page is blank.)

TABLE OF CONTENTS

TABLE OF CONTENTS

(continued)

TABLE OF CONTENTS
(Concluded)

LIST OF FIGURES

LIST OF FIGURES
(Continued)

LIST OF FIGURES
(Continued)

LIST OF FIGURES
(Continued)

LIST OF FIGURES
(Continued)

LIST OF FIGURES
(Continued)

LIST OF FIGURES
(Continued)

LIST OF FIGURES
(Continued)

LIST OF FIGURES
(Continued)

LIST OF FIGURES
(Concluded)

LIST OF TABLES

LIST OF TABLES
(Continued)

LIST OF TABLES
(Continued)

LIST OF TABLES
(Concluded)

(The reverse of this page is blank.)

SECTION I

INTRODUCTION

A. OBJECTIVES

Because the Air Force is the primary user of the rocket fuels, hydrazine (Hz), monomethylhydrazine (MMH), and 1,1-dimethylhydrazine (UDMH), it is responsible for the environmental implications associated with the transport, storage, and handling of these fuels. During handling, hydrazine fuels could inadvertently be released to the atmosphere and the surrounding aqueous and terrestrial environments. Therefore it is advantageous to understand the fate of these materials in soil and water. The purpose of Task Order 85-4 was to study the soil chemistry, soil microbiology and transport through the soil of the hydrazines. The objectives of the research conducted under Task Order 85-4 were as follows:

(1) Conduct a literature review of all pertinent facets of interactions of hydrazine fuels in the soil/water environment.

(2) Investigate the decomposition of hydrazine fuels in aqueous media under conditions of varying chemical compositions, pH, redox potential, ionic strength, and temperature.

(3) Describe the adsorption/desorption characteristics of hydrazine fuels on various soils and soil components. Investigate, on a molecular level, the nature of hydrazine and clay mineral interactions.

(4) Study the effects of hydrazine fuels on microbial population dynamics, species survival, soil respiration, and microbial degradation of hydrazine.

(5) Conduct soil column experiments to evaluate the transport of hydrazine fuels through soils. Develop breakthrough curves and transport model parameters for prediction of hydrazine movement in soils.

The literature review has been compiled and submitted as a separate publication. The initial objectives included research with hydrazine, monomethylhydrazine, and 1,1-dimethylhydrazine; however, after beginning

the research project it was mutually agreed to focus on hydrazine alone. Some microbiological experiments were conducted with MMH and are reported in Section IV.

B. BACKGROUND

1. Water, Soil, and Soil Constitutent Studies

The aqueous chemistry of hydrazine in both deionized and distilled water and natural waters was investigated under the best available laboratory techniques and equipment. The soil and clay adsorption studies were conducted in a modified "batch slurry" technique similar to the flow-through system described by Hayes et al. (Reference 1). The clay surface studies, microbial experiments, and the transport model development are described in the separate chapters on each of these areas.

2. Spectroscopic Investigations

In situ vibrational spectroscopic methods have been used to study the interaction of hydrazine with kaolinite surface. Noninvasive Raman and FT-IR spectra were obtained for the kaolinite-hydrazine (KH) intercalation complex at low temperatures and pressures. Strong vibrational perturbations of the guest intercalate, hydrazine, and of the clay mineral kaolinite were observed. Upon expansion of the kaolinite structure by hydrazine, a significant reduction in the intensities of the inner-surface hydroxyl groups was observed which indicated that strong hydrogen bonds were formed between the intercalated hydrazine species and the kaolinite interlamellar surface. In addition, several Raman- and IR-active vibrational modes of hydrazine were influenced by the kaolinite surface.

3. Soil Microbiology

Information on the degradation and soil microbiology of hydrazine and the substituted hydrazines is limited. Experimental procedures

for evaluating the fate and transformation of the hydrazine fuels were developed using standard soil microbiological techniques. The effects of Hz and MMH on soil microbiological properties were also examined.

4. Soil Physics and Model Development

A brief literature search indicated a general deficiency of experimental information concerning the mobility and transport of hydrazine in soils. Such information is important to any evaluation of the potential for chemical contamination of groundwater in the event of accidental spillage of hydrazine onto a soil surface or of subsurface leakage of hydrazine from defective storage tanks. Any contamination of groundwater should be considered detrimental to environmental quality since hydrazine is an "animal carcinogen" as well as a "suspected human carcinogen."

Research was initiated to determine the mobility of hydrazinium (ionic form of hydrazine) during steady flow through columns of water-saturated sandy soil and to develop a simplified numerical mathematical model to describe the resulting transport of hydrazinium through the soil. Factors investigated include the concentration of hydrazinium in applied aqueous solutions, pore water velocity during flow, and the method (pulse versus continuous) of solution application. Material from three horizons (A_p E2 and E2) of an Arredondo fine sand soil were used in the hand-packed soil columns.

A finite-difference numerical model was also developed to describe 2-dimensional transport of hydrazine during transient water flow in a rectangular soil region that receives rainfall input at the soil surface, subsurface leakage of hydrazinium solution at the upstream boundary, and drainage of groundwater at the downstream boundary. An impervious clay layer was assumed to provide the lower boundary of the flow region.

C. SCOPE

The scope of this report is to document the results of various soil chemistry, soil microbiology, and soil physic laboratory experiments such that a fate and transport model could be evaluated. The approach was to intergrate the results from several aqueous and soils experiments to better understand the chemical behavior of hydrazine in natural waters and soil environments. A parallel approach for the effects of hydrazine and monomethylhydrazine on soil microbial processes was also taken.

Information obtained from the soil chemistry and soil microbiology studies was utilized in the development of a 1- and 2-dimensional transport model for evaluating the potential for hydrazine to migrate in soils. The development of such a fate and transport model would allow the U.S. Air Force an accessment tool for taking remedial action in the events of accidential spills or leaks from storage tanks.

SECTION II

AQUEOUS AND SOIL CHEMISTRY STUDIES

A. MATERIALS AND METHODS

1. Reagents

Hydrazine monohydrate was purchased from Aldrich Chemical Company, (Milwaukee, WI.). All other chemicals were analytical grade or the highest grade available.

2. Natural Waters

Natural waters were collected in 20-gallon plastic carboys from the Saint Johns River and Santa Fe Lake. River water samples were collected under the East Palatka Bridge located on US Highway 17. The lake samples were taken near Buddy's Landing on Lake Santa Fe in Melrose, Florida. After arrival to the laboratory the dissolved oxygen concenration and total carbon content were measured. Two liters of each type of water were autoclaved for 20 minutes and 2 liters were filtered through a 0.2 μm membrane to sterilize and eliminate solids in suspension. They were stored in a refrigerator at 10°C until ready for use.

3. Clays

Montmorillonite (SAz-1) and kaolinite (KGa-1) used in these studies were obtained from the Clay Minerals Repository, Department of Geology, University of Missouri.

4. Soils

The three top horizons of an Arredondo fine sand (grossarenic paleudult, loamy, siliceous, hyperthermic): Ap (0-20 cm), El (20-80) cm)

and E2 (80-150 cm), were collected from a Northwest location in Alachua County, Florida.

The Orangeburg loamy sand soil (Typical Paleudult, fine-loamy, siliceous, thermic) was a sample from the top 15 cm taken from a site in Jackson County, Florida.

5. Analytical Methods

The laboratory method selected to determine hydrazine concentrations less than 11 mmol l^{-1} was a modification of the procedure used by Hayes et al. (Reference 1). Our technique substituted hydrochloric acid for trichloroacetic acid, since only aqueous solutions were analyzed. Higher concentrations of hydrazine were analyzed by direct iodate titration using carbon tetrachloride to detect the end point (Reference 2).

The concentration of oxygen in aqueous solutions contained in small bottles was determined with a dissolved oxygen microelectrode (Microelectrodes, Inc.). For the studies in a Pyrex® cell a dissolved oxygen electrode Orion model 97-08, was used. Total metals concentrations were determined on an atomic adsorption spectrophotometer (Perkin Elmer 460). A specific gas-sensing electrode (Orion 951201) was used to detect ammonia in the degradation studies.

The technique used to determine Cu^{+2} in the supernatant of clays was differential pulse stripping voltametry measured with a EG&G Princeton Applied Research polarographic analyzer model 384B. The supporting electrolyte was NH_4^+-citrate at pH 3. Because this technique does not distinguish between Cu^{+2} and Cu^{+1}, a specific ion electrode was used to measure Cu^{+2} whenever hydrazine was present.

The concentration of total soluble organic carbon in water was determined on a Total Carbon System from Oceanography International Company, Model 0524B. A small aliquot was placed in a glass ampule with a series of reagents to acidify and digest the organic forms present. Ampules were purged of inorganic carbon with purified oxygen. After ampules were sealed, they were autoclaved to convert the organic carbon to CO_2 which was measured by a nondispersive infrared analyzer equipped

with a digital integrator. The integrated peak area was related to the weight of carbon by comparison with standards of known carbon content.

B. EXPERIMENTAL PROCEDURES

1. Aqueous Studies

Solutions of hydrazine ranging from 10 to 500 μg l^{-1} were prepared in 2 liter volumetric flasks. After the desired chemical compositions were obtained, 10 mL aliquots were put into small bottles and incubated in a constant-temperature water bath. Three bottles were opened for analysis at various time intervals. The frequency of analysis depended on the rate of hydrazine degradation. Various constant ionic strengths were maintained using $CaCl_2$. Acidic pH's were obtained with HCl. Neutral pHs were obtained with phosphate buffers at different ionic strengths.

Three different types of bottles were used in the study to investigate the effects of container material: glass scintillation vials made of borosilicate low in potassium content, glass serum vials made of borosilicate (with crimped aluminum tops and a Teflon® liner), and polyethylene bottles. Most of the experiments were carried out in serum vials, assumed impermeable to oxygen. New bottles were opened each time an analysis was needed and the rest of the solution was discarded. However, in a recent experiment we found that oxygen could diffuse slowly into the bottles. Because of this, we incubated vials in an anaerobic incubator under a nitrogen atmosphere to minimize autoxidation.

A study to determine the effect of the container size on hydrazine degradation used serum vials of 10, 50 and 100 mL. Vials were opened periodically for analysis, shaken and put back in the constant temperature water bath for future analysis.

Two experiments were were conducted to determine effect of the gas atmosphere on hydrazine degradation. Solutions were prepared by flushing the stock hydrazine solutions with nitrogen or argon gases. Vials were also flushed with the gases during the transfer of 10 mL of hydrazine solution until they were sealed. This procedure proved to be

inadequate because some air entered during the process of closing the vials and it was later discovered that the seal was not completely impermeable to oxygen.

One set of experiments was carried out in a 1.5-liters Pyrex® cell with a glass top, to which a pH electrode, an oxygen electrode and a redox electrode were attached. The cell also had an entrance for bubbling gases and a syringe to withdraw samples. The contents of the cell were continuously stirred with a Teflon®-coated magnetic stirring rod. Different Cu^{+2} concentration solutions were equilibrated with gases of a known oxygen content (0.27 μmol l^{-1} or less). Under ambient air conditions a trap was installed between the air pump and the cell to remove CO_2. It was found later that the CO_2 trap had a slight effect on the pH of the experimental solution. After the addition of 10.3 mmol l^{-1} of hydrazine the electrode readings were monitored and samples were withdrawn periodically for hydrazine analysis.

2. Clay Studies - Hydrazine Degradation Studies on Cu-Montmorillonite

Clay fractions smaller than 0.5 μm were selected for the degradation studies. This was accomplished by washing the clay three times with 0.1 N NaCl to disperse the particles and saturate the exchange complex with a single cation. Samples were washed with deionized water until a negative chloride test resulted. The smaller fractions were separated using an ultraspeed centrifuge. To minimize dissolution of clays, samples were stored in 0.1 N NaCl solution until ready for use.

In the degradation studies 1 mL aliquots of different Cu^{+2} stock solutions were added to serum vials containing 9 mL of Na-saturated montmorillonite suspension (3 mg ml^{-1}). Vials were equilibrated for 3 days and Cu^{+2} concentration in the supernatant was measured by differential pulse-stripping voltametry. Hydrazine was added to the vials and hydrazine concentrations in the suspension were measured with time. The standard curve was developed, adding the same amount of clay to the volumetric flask as to the volumetrics containing the experimental samples. After centrifugation of the serum vials, hydrazine was measured in the

supernatant. The difference between the amount of hydrazine in the suspension and in the supernatant was assumed to be adsorbed on the surface of the clay.

Initially an isotherm was run in a Na-montmorillonite suspension prepared in the same was as the suspension for the degradation studies described above. Ten-milliliter aliquots containing 3 mg ml^{-1} of clay were placed in serum vials under anaerobic conditions. Increasing amounts of hydrazine solution were added to the vials. Afterwards vials were stoppered, shaken and let equilibrate in the absence of oxygen for 24 hours. The pH of the suspension was not controlled and it increased, depending on the concentration of hydrazine in the supernatant. Hydrazine concentration left in the supernatant was measured and adsorption was calculated as the difference between the amount of hydrazine added and the amount remaining in the supernatant.

Adsorption isotherms of kaolinite and montmorillonite were measured at pH 4 and 8. Four grams of kaolinite and 1 gram of montmorillonite without any pretreatment were placed in preweighted polyethylene centrifuge bottles. The contents were washed five times with 0.1 N NaCl at the pH of the experiment to saturate the exchange complex with Na^{+1}. All the steps except centrifuging were carried out in an anaerobic incubator to minimize autoxidation. Afterwards the excess of NaCl in the supernatant was washed three times with deionized water at the pH of the suspension. The supernatants from last wash were kept for analysis of Fe, Si, Na, and pH. The volumes were brought to 20 mL with deionized water and increasing amounts of hydrazine solution were added to the centrifuge tubes. Samples were shaken and left equilibrating in the anaerobic incubator for 24 hours. After centrifuging hydrazine, Na, Fe, and Si concentrations were measured in the supernatant. Hydrazine adsorbed was calculated as the difference between the amount of hydrazine added and the amount left in the supernatant after equilibration with the suspension. The difference between the amount of sodium in the supernatant after equilibration and before adding hydrazine is an indication of the amount of hydrazine exchanged with Na^{+1} on the exchange complex. All samples were analyzed for Fe and Si after early analysis indicated excessive Na in solution was probably due to clay decomposition. After the

completion of the isotherms the samples with the highest concentration of hydrazine were washed exhaustively with 0.1 N KCl to desorb hydrazine from the clays.

3. Soil Studies

Samples from the three top horizons of Arredondo soil were air-dried and sieved through a 2 mm sieve. Metals were analyzed in a double acid extract. Two sets of isotherms were conducted. The first set was done at pH 4.8 and 8.0, maintaining a constant ionic strength of 0.01 N with $CaCl_2$ solutions. We lowered the pH of the hydrazine stock solutions with HCl. Because $N_2H_5^+$ Cl^- contributes to the salt content of the solution, the concentration of hydrazinium chloride was considerated in preparing the constant ionic strength solutions. The isotherms were obtained over a wide range of concentrations. Five-gram samples of each horizon of Arredondo soil were placed in glass serum vials and 10 mL of hydrazine solution with constant ionic strength, and increasing hydrazine concentrations were added. After incubation for 48 hours under anaerobic conditions, samples were centrifuged and hydrazine was measured in the supernatant.

The second set of isotherms was conducted at pH 4 and 8 in the anaerobic incubator. Twenty grams of soil were washed five times with 0.1 N NaCl at the pH of the study to saturate the exchange complex with a single cation. Afterwards samples were washed with deionized water at the same pH to eliminate the excess NaCl in the supernatant. The supernatants of last wash were kept for analysis of Fe, Si, Na, and pH. The volumes were brought to 20 mL with deionized water and increasing amounts of a hydrazine stock solution were added to the centrifuge tubes. Hydrazine were measured and calculated as in the procedure for kaolinite and montmorillonite. After extraction of hydrazine with 0.1 N KCl, soils were extracted twice more with dilute acid.

Degradation of hydrazine in the presence of Orangeburg soil was done as follows: One, 0.5, and 0.25 grams of Orangeburg soil were placed in serum vials and 10 mL of 10.3 mmol l^{-1} hydrazine solution were added to each vial. Hydrazine solution in the suspension was measured with

TABLE 1. CHEMICAL PROPERTIES OF NATURAL WATERS.

CONCENTRATIONS

($mg\ l^{-1}$)

	Sta Fe Lake	St. Johns River
pH (s.u.)*	6.2	8.0
O_2	6.7	7.0
Organic C	6.23	7.16
K	0.4	5.7
Na	5.9	142
Ca	2.5	59
Mg	1.4	23
Cu	0.00	0.00
Fe	0.0	0.0
Mn	0.00	0.00
Zn	0.01	0.01
NH^{+4}	0.2	0.01
NO_3^-	0.01	0.1
P	T**	T
Cl^-	12	320

* (s.u.) standard units

** T traces

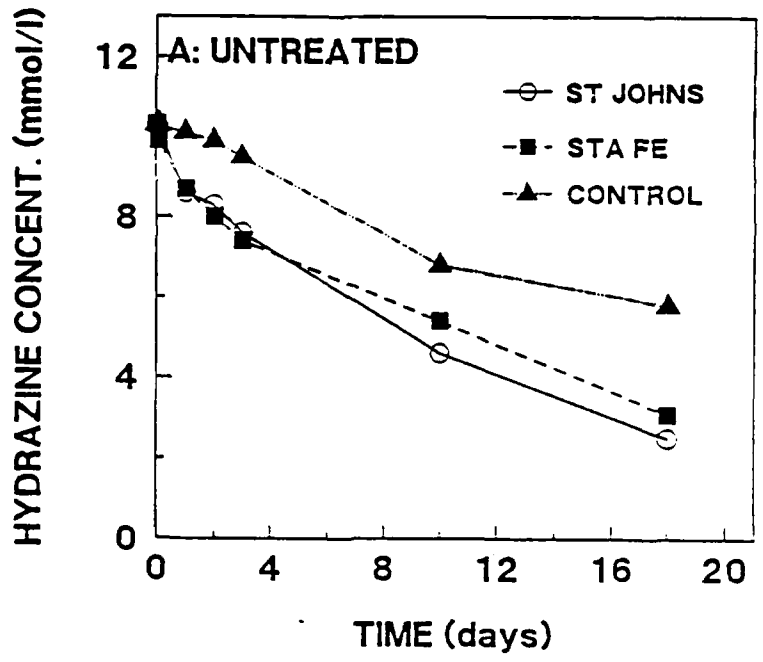
time. After centrifugation of the vials, hydrazine was measured in the clear supernatant.

C. RESULTS AND DISCUSSION

1. Degradation of Hydrazine in Aqueous Systems

a. System I - Natural Waters

Selected chemical properties of the two natural waters,



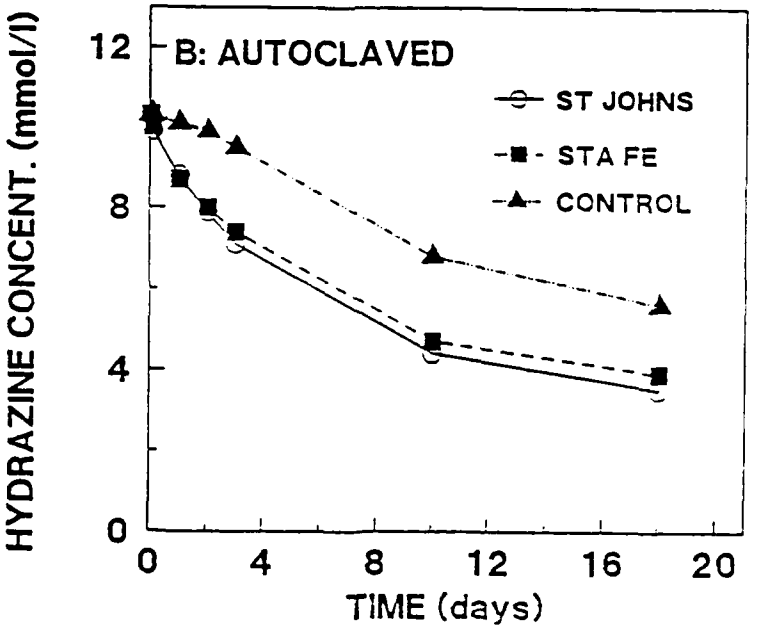


Figure 1. Degradation of Hydrazine in Natural Waters (at 22°C in Serum Vials): Effect of Sterilization; (A) Untreated, (3) Autoclaved.

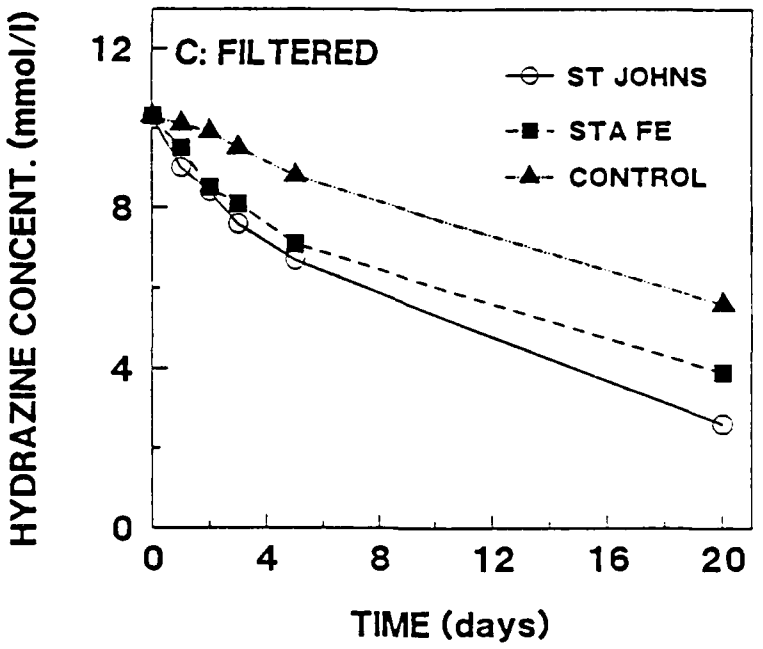


Figure 1. Degradation of Hydrazine in Natural Waters (at 22°C in Serum Vials): Effect of Sterilization; (A) Untreated, (B) Autoclaved.

Saint Johns River and Santa Fe Lake are presented in Table 1. The primary differences between these two natural waters were a higher pH and a higher salt content in the Saint Johns River water. Upon addition of 10.3 mmol l^{-1} of hydrazine, the pH of the solutions increased to 10.2. Degradation was slightly faster in the river water than in the lake water, and in natural waters degradation was faster than in the distilled water control (Figure 1). The experiment was conducted in serum vials at the original oxygen content and under a nitrogen atmosphere. Because there was no significant difference between either atmosphere we concluded that oxygen was leaking into the serum vials.

As shown in Figure 1, sterilization had very little effect on the rate of hydrazine disappearance, indicating that the mechanism was mainly chemical and not microbiological. This is in agreement with the findings by Ou and Street (Reference 3) that concentrations above 5 mmol l^{-1} might be toxic for microbial populations.

b. System II - Distilled Water

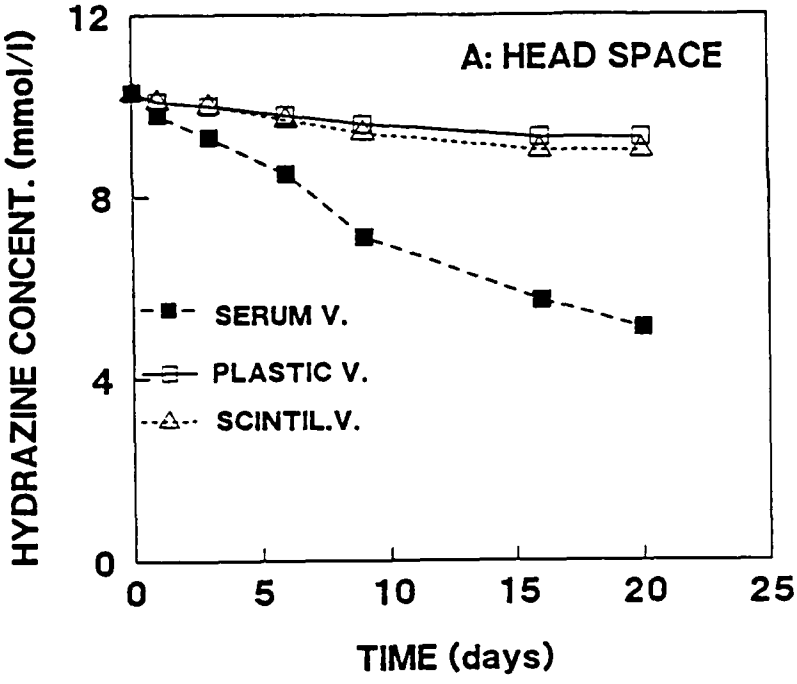
(1) Effects of Reaction Vessels

It has been previously reported that the chemical composition of the reaction vessel might have an effect on the degradation rate of hydrazine. The first experiment used hydrazine solutions prepared in distilled water at 22°C, and compared three types of containers: glass serum vials, polyethylene vials, and glass scintillation vials. Each type of container underwent four treatments:

under air atmosphere, with head space;
under nitrogen atmosphere, with head space;
under air atmosphere, no head space;
under nitrogen atmosphere, no head space.

Degradation was slow in both the polyethylene and the scintillation vials, with no significant treatment effect for either vial (Figure 2). After 24 days only 9.5 percent of the added hydrazine had degraded. This suggests that these kind of materials had no catalyzing effect on the decomposition of hydrazine. The solution pH remained at 10.2 for the nitrogen treatment and 9.9 for the air treatment during the course of the incubation. The fact that the nitrogen treatments increased their oxygen content with time suggests that the sealing in these kind of bottles was not impermeable to oxygen.

On the other hand the serum vials had an important effect on hydrazine degradation. Degradation was faster in the vials that had air in the headspace because oxygen diffused freely into the solution. For the other treatments oxygen had to diffuse first through



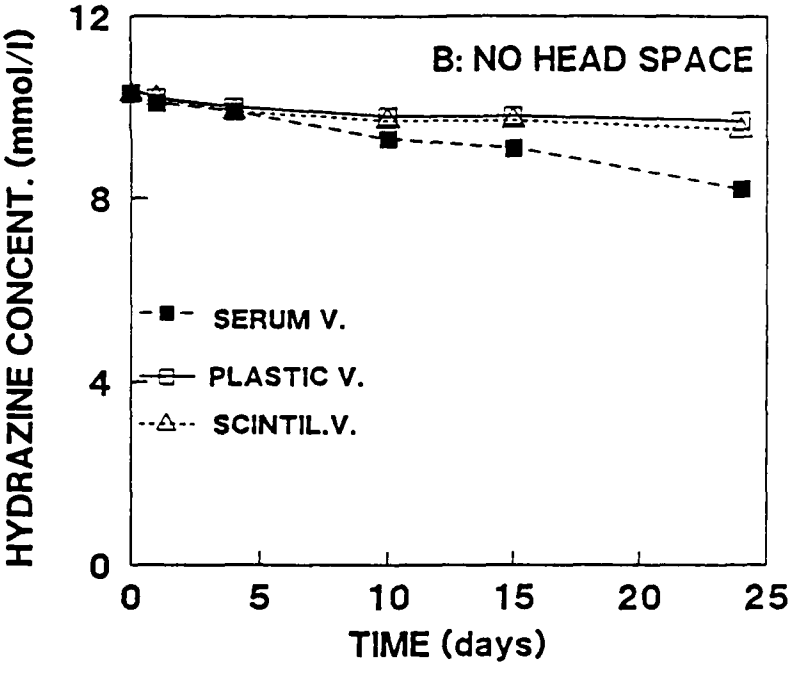


Figure 2. Hydrazine Degradation in Aerated Distilled Water at 22°C: Effect of Container Material; (A) Headspace Filled With Air (B) Bottles Filled to the Top With Solution.

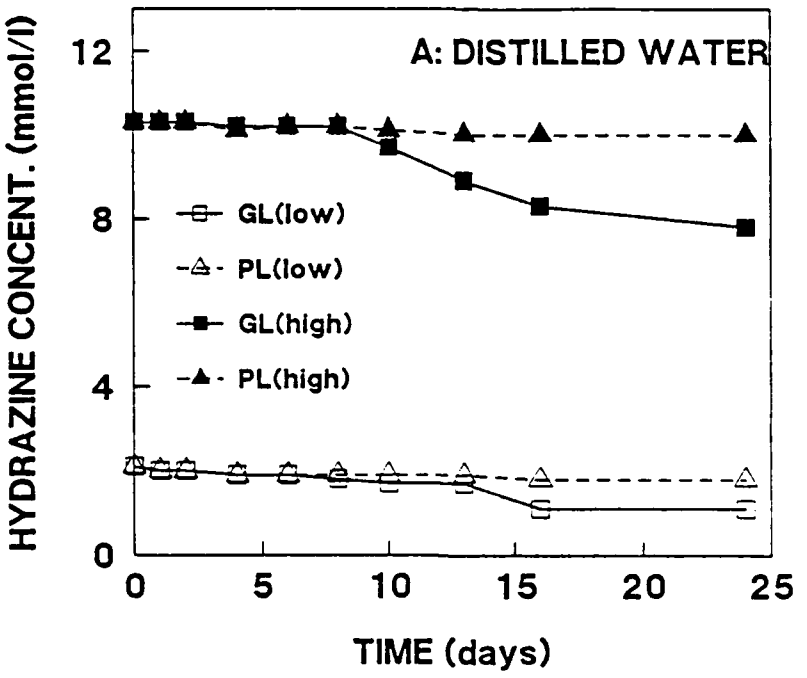
the seal, consequently hydrazine degraded slower. The oxygen content in the vials with no head space remained nearly constant at 1 mg l^{-1}.

In another set of experiments at 45°C and in the presence of Cu^{+2} degradation was faster in polyethylene bottles than in glass serum vials (Figure 3). Initially this was attributed to faster leakage of oxygen into the polyethylene bottles but this may not be conclusive since the rate of oxygen replenishment in both types of bottles after hydrazine had disappeared was similar.

Ammonia was present as a degradation product only in the solutions containing Cu (Figure 4a). A lineal relationship was found between hydrazine degradation and ammonia production at 25°C. Oxygen concentration in the solutions containing Cu stayed close to 0 ppm as long as some hydrazine remained. After hydrazine had degraded completely, the oxygen levels in solution increased until reaching equilibrium with air (Figure 5a). The rate of oxygen replenishment was similar in the serum vials and the polyethylene bottles, further confirming that the seals in both types of containers were not impermeable to oxygen.

To further investigate the effect of serum vials on the stability of hydrazine an experiment was set up using three different sizes of the same type of serum vial with exactly the same sealing system. Hydrazine solutions were prepared in five different treatments: distilled water, 0.001 ppm Cu, 0.01 ppm Cu, 0.1 ppm Cu, and 1.0 ppm Cu and they were incubated at 45°C. This set of vials was opened for analysis and sealed again after air had filled the head space. In this case diffusion of oxygen through the seal was not the controlling factor. Data from this experiments are shown in Table 2.

The size of the container had no effect on the degradation of hydrazine in distilled water (Figure 6a). However, in the presence of Cu hydrazine degraded faster in the 10 mL vials that had a much larger headspace to solution ratio and surface to volume ratio than the 50 or 100 mL vials (Figures 6c,d,e). Comparisons of other two vial sizes indicated hydrazine degraded slightly faster in the 50 mL vials (Figures 6d,e). Both the 50 mL and 100 mL vials had the same headspace to solution ratio but the smaller one had a slightly higher amount of solution in contact with the wall of the container. The results indicate there is



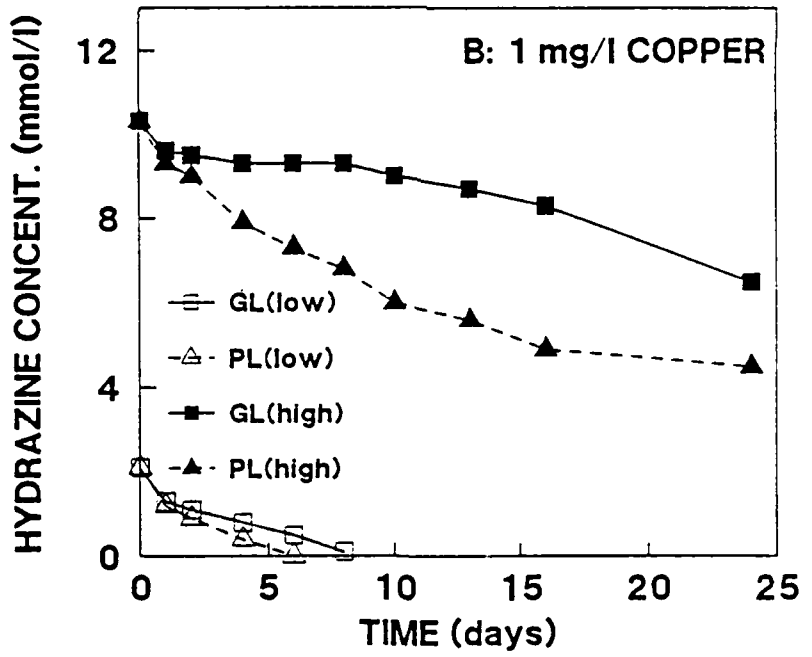
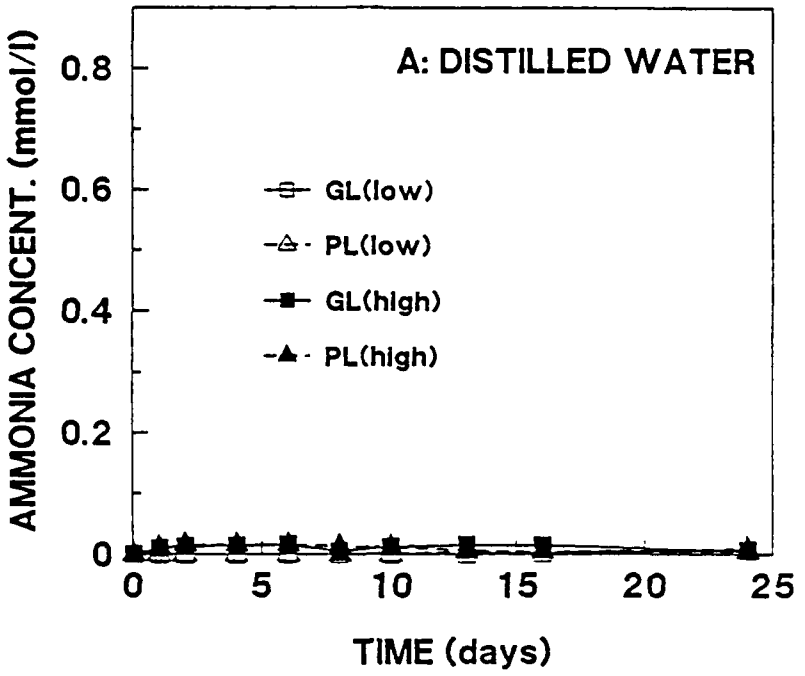


Figure 3. Hydrazine Degradation at 25°C. Effect of Container Material: GL-Serum Vial, PL-Polyethylene Vial, (Low)-Low Initial Hydrazine Concentration, (High)-High Initial Hydrazine Concentration; (A) in Distilled Water, (B) in the Presence of CU.



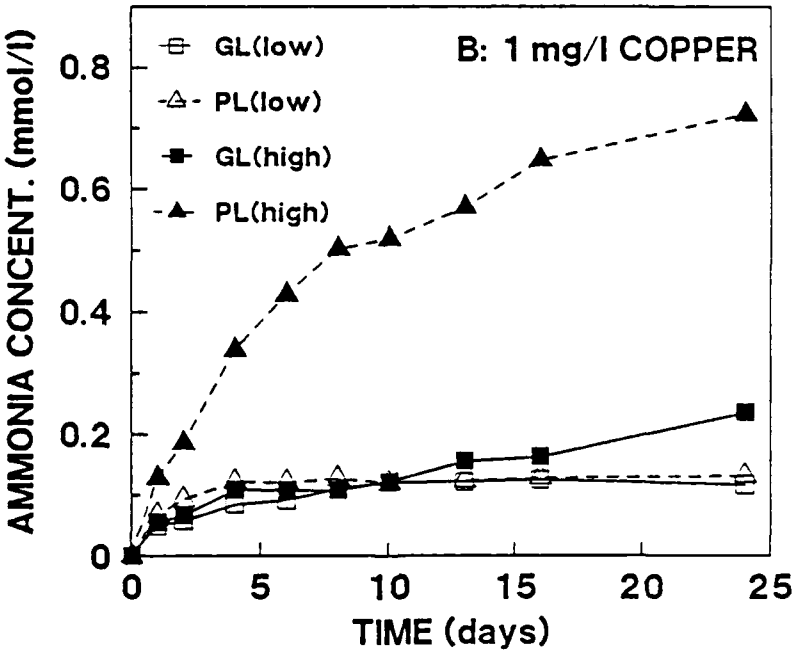


Figure 4. Ammonia Evolution From Hydrazine Degradation at 25°C. Effect of Container Material (Legend as in Figure 3); (A) in Distilled Water, (B) in the Presence of Cu.

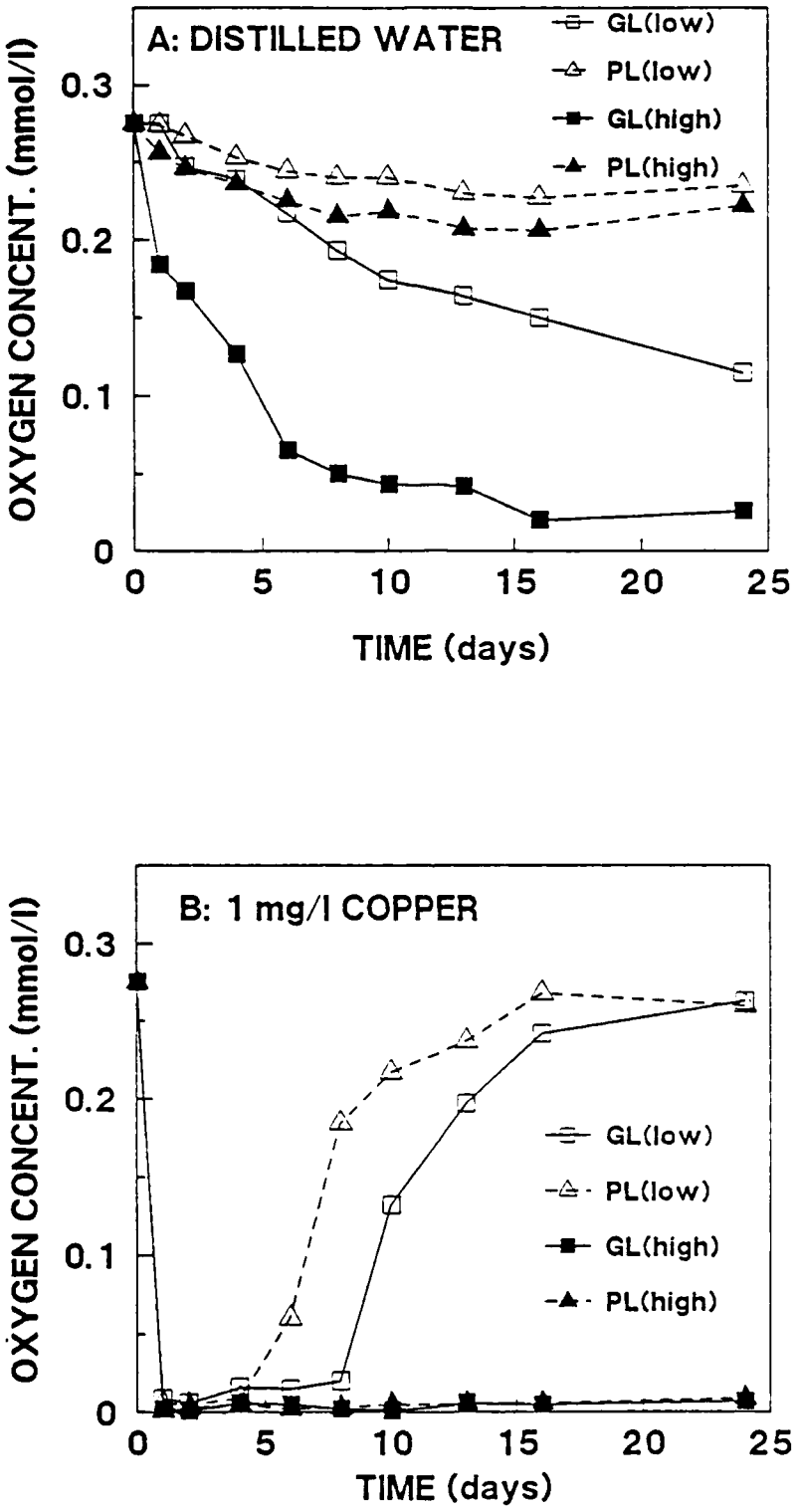
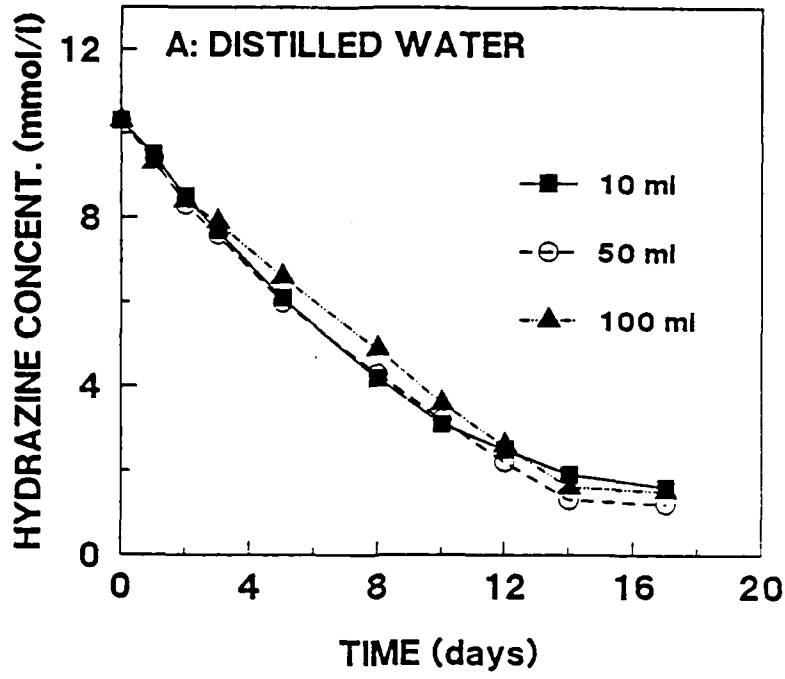


Figure 5. Changes in Oxygen Concentration During Hydrazine Degradation at 25°C. Effect of Container Material (Legend as in Figure 3); (A) in Distilled Water, (B) in the Presence of Cu.



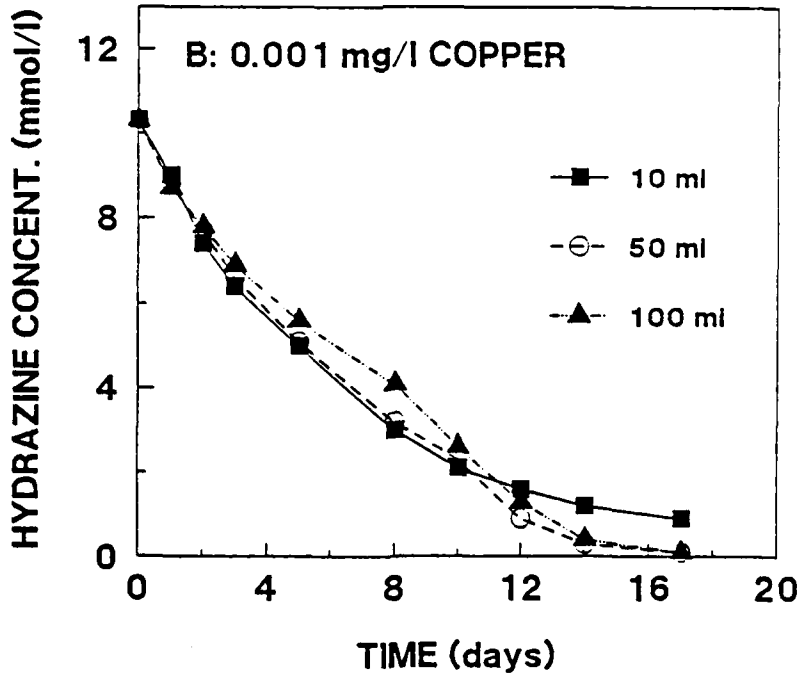
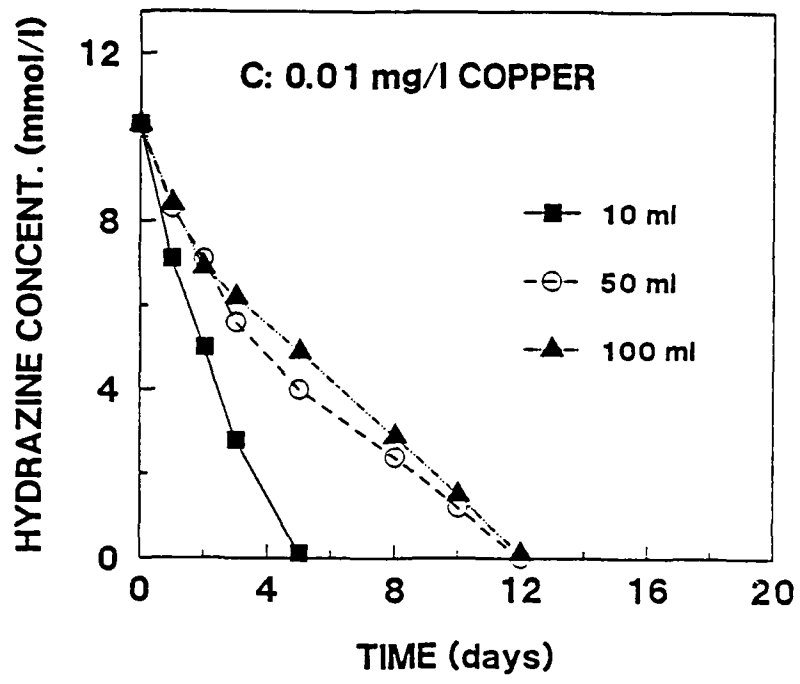


Figure 6. Degradation of Hydrazine at 45°C. Effect of Container Size; (A) in 0.001 ppm Cu Solution, (B) in Distilled Water.



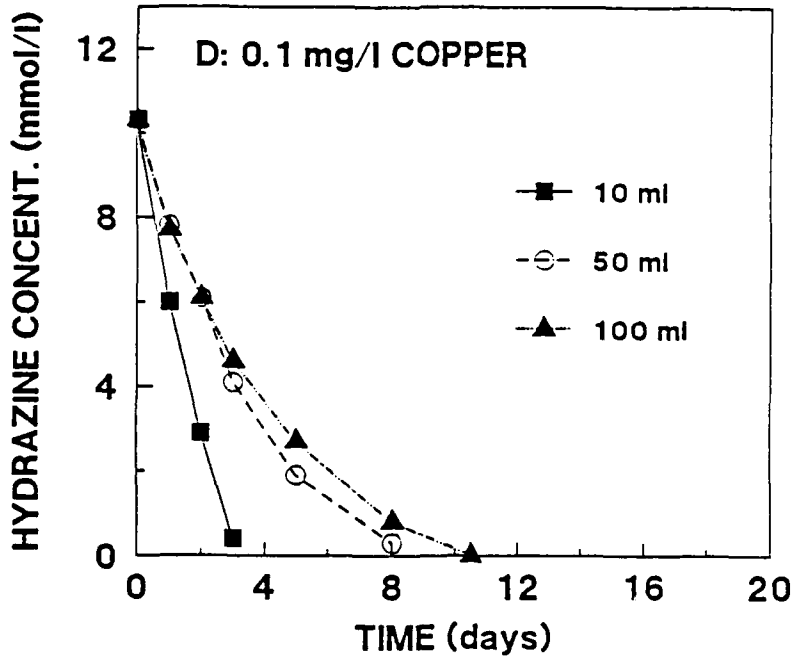


Figure 6. Degradation of Hydrazine at 45°C. Effect of Container Size; (C) in 0.001 ppm Cu solution and (D) in 0.1 ppm Cu solution.

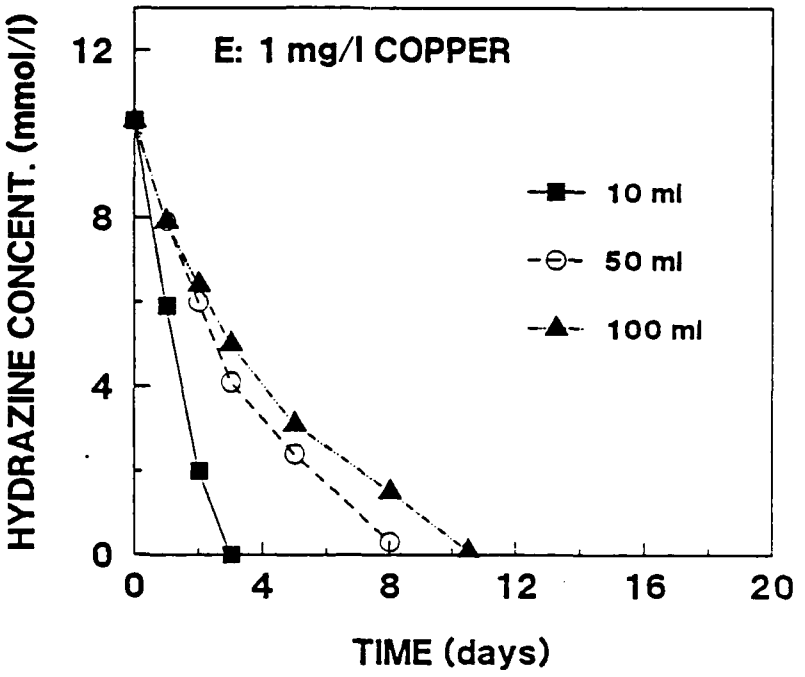


Figure 6. Degradation of Hydrazine at 45°C. Effect of Container Size; (E) in 1 ppm Cu Solution. (Concluded)

TABLE 2. AVERAGE MEASUREMENTS OF THE SERUM VIALS.

	10 mL	50 mL	100 mL
headspace (ml)	4.4	10.2	20.6
height of solution (cm)	2.90	4.40	6.00
internal radius (cm)	1.05	1.90	2.30
gas-liquid interphase (cm^2)	3.46	11.3	16.6
head space/solution (ml/ml)	0.44	0.20	0.20
contact surface/volume (cm^2/ml)	2.25	1.28	1.03

an interaction between the Cu^{+2} ions and the hydrazine on the surface of the container. This interaction was more pronounced as the solution Cu^{+2} concentration increased.

(2) Effect of Solution Ionic Strength

The experiment to evaluate the effect of ionic strength on hydrazine degradation was carried out in scintillation vials at 22°C. The ionic strength was adjusted using $CaCl_2$. As shown in Figure 7, ionic strength had a significant effect on the slope of the degradation curve. Increasing the $CaCl_2$ concentration increased the rate of hydrazine autoxidation. The effect of water hardness on hydrazine degradation has been previously reported (Reference 4).

(3) Effect of pH

In preliminary studies conducted over a wide range of pHs the fastest rate of hydrazine degradation was found at pH 7.0 (Figure 8a). These results contradicted what some authors have reported previously (Reference 5). For this reason, a study was set up, maintaining a pH of 7.0, but reducing the ionic strength of the phosphate buffering system. Hydrazine degradation was approximately first order with respect to phosphate concentration (Figure 8b). This result implied that phosphate ion concentration and not pH was the primary reason for the increased degradation rate of pH 7. When the reaction was carried out in scintillation vials at 22°C, degradation was very slow for both pH 7 and 9.5 solutions (Figure 9). The ionic strength of the pH 7.0 buffer was 0.1, but in this case the buffer catalysis was not as important as in the serum vials. This suggests an interaction between the phosphate and the wall of the container as previously reported for Cu^{+2}.

In the series of reactions carried out in the 1.5-liter Pyrex® cell it was not intended to evaluate the effect of pH. However, to remove CO_2 from the air we used a NaOH trap. The NaOH trap had a small effect on the pH of the hydrazine solution in the presence of 1 $\mu g\ l^{-1}$ of Cu ion and it resulted in an increase in autoxidation of hydrazine

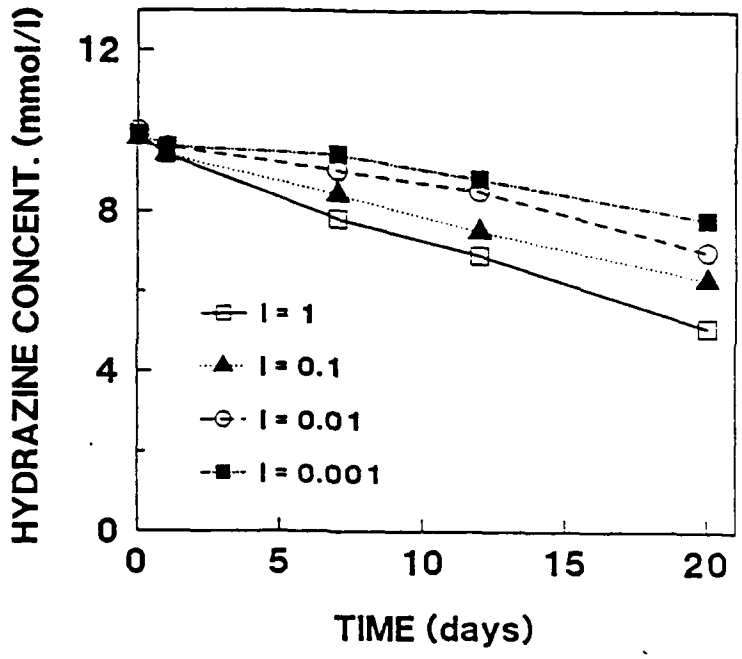


Figure 7. Hydrazine Degradation at 22°C in Scintillation Vials: Effect of Solution Ionic Strength (I).

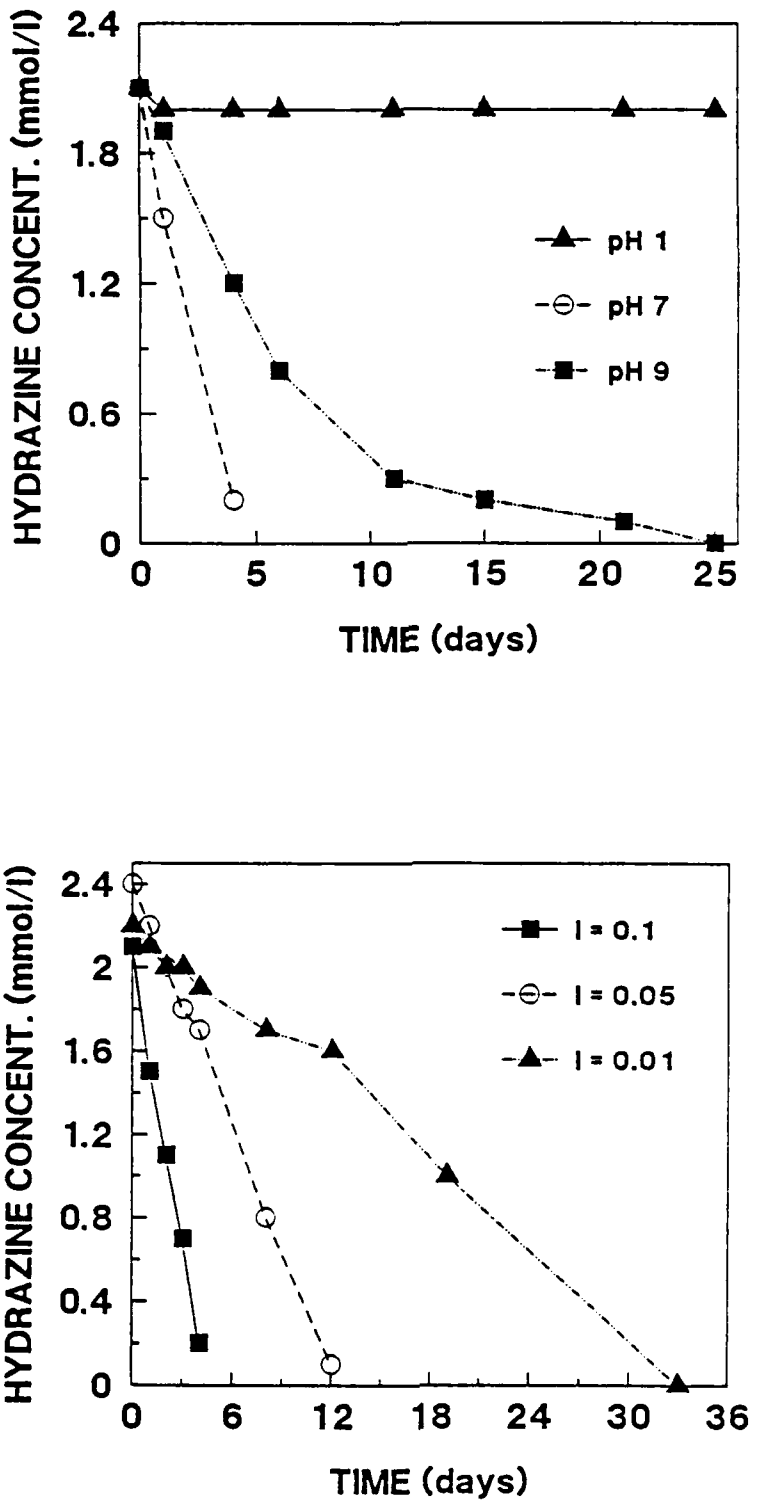


Figure 8. Hydrazine Degradation at 45°C in Serum Vials: (A) Effect of pH, (B) Effect of Phosphate Buffer Ionic Strength at pH 7.

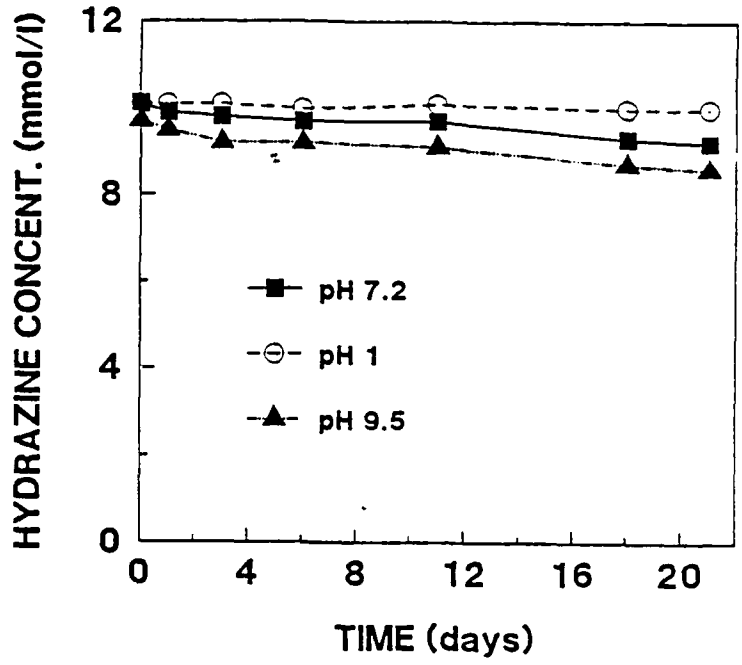


Figure 9. Hydrazine Degradation at 22°C in Scintillation Vials: Effect of pH

above pH 9.4. Hydrazine degradation was negligible below pH 4 and stopped completely at pH 2.

2. Effect of Gas Atmosphere

Our objective in preliminary studies was to evaluate the effect of the composition of the gaseous atmosphere on hydrazine decomposition in aqueous systems. To achieve this objective we conducted two experiments at 22°C and 45°C using air, 100% nitrogen and 100% argon atmospheres. Even though we found differences among gases (Figure 10) we think the data is erroneous since the vials were proven to be slightly permeable to air. When the experiment was conducted in the Pyrex reaction cell, bubbling nitrogen gas throughout the experiment resulted in no hydrazine degradation after 24 days.

3. Effect of Cu

The catalytic effect of Cu^{+2} on hydrazine autoxidation has been reported (References 5,6,7). However those studies were carried out at

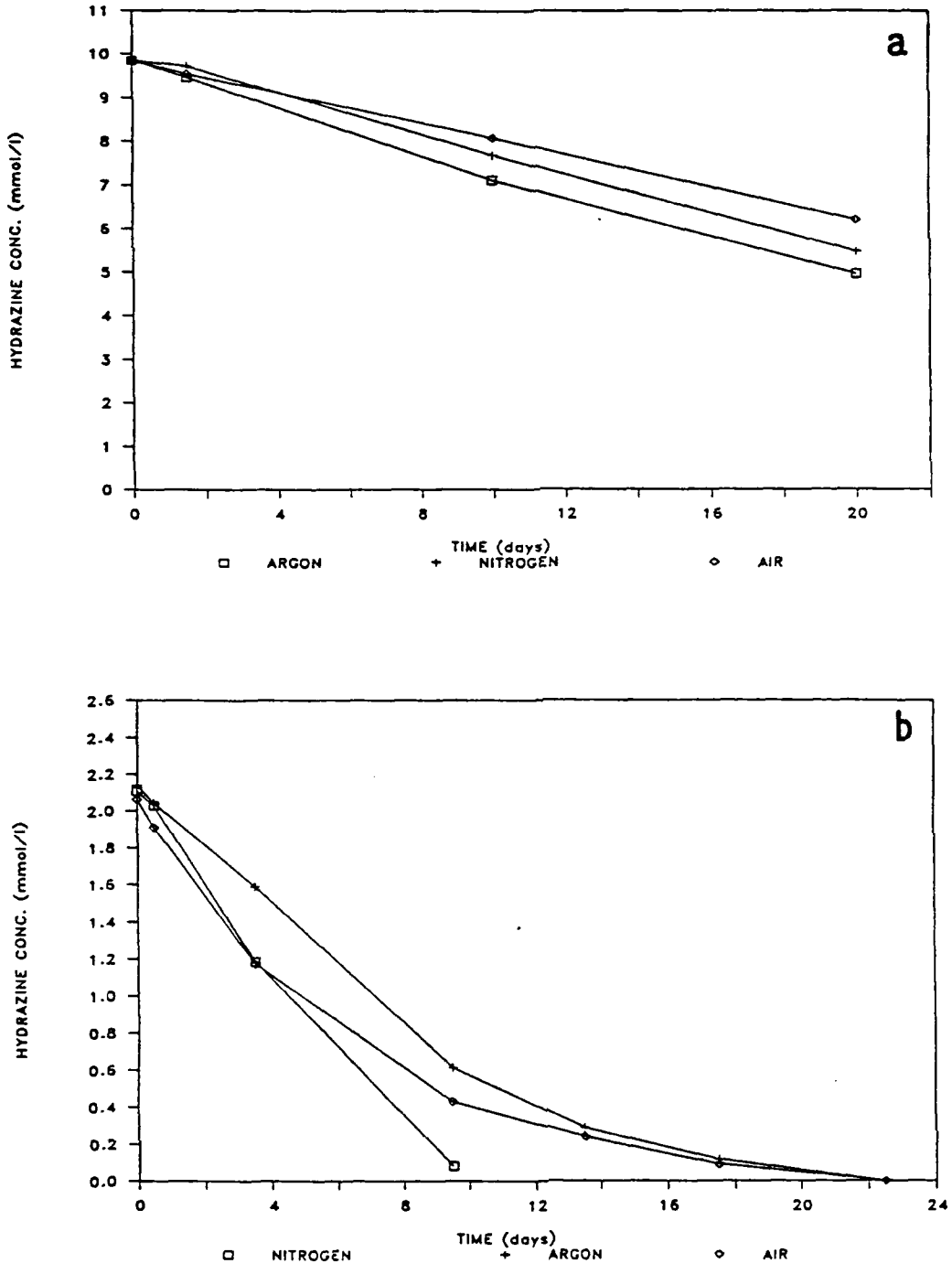


Figure 10. Hydrazine Degradation in Serum Vials: Effect of Gas Atmosphere; (A) 22°C, (B) 45°C

pH 12, where the hydrazine degradation rate is a maximum (Reference 7). To simulate what happens in the soil environment we found necessary to carry the experiments at the pH obtained by the dissolution of hydrazine in water without further adjustments, and with Cu concentrations similar to those found in soils. In every case results were compared with those obtained in deionized-bidestilled water whose Cu concentration was low enough to be undetected by differential pulse stripping voltametry (<0.1 p.p.b.).

The first experiment at 25°C hydrazine did not degrade in the absence of Cu in the polyethylene bottles (Figure 3b). However, in the serum vials, autoxidation occurred after an induction period. This result agrees with Audrieth and Mohr's findings at a higher pH.

On the other hand, when 1 ppm Cu was present, hydrazine degraded faster in the polyethylene bottles than in the serum vials (Figure 3a). Visual observation of the experimental solutions containing Cu revealed the presence of a reddish precipitate in the low hydrazine concentration solutions (2 mM), and an almost black precipitate in the high hydrazine concentration solutions (10.3 mM). The precipitates could not be isolated because of their instability in the presence of air.

When oxygen was in excess over hydrazine (Figure 11a), degradation was initially fast and the stoichiometry of the reaction was 1:1. Ammonia, if produced, was below the detection limits of our technic. After the initial degradation, hydrazine concentration changed very slowly although the solution was saturated with oxygen (Figure 11b). Ten percent of the initial hydrazine concentration remained after 60 days.

Increasing the temperature to 30°C and 45°C accelerated the rate of initial reaction. However, after the oxygen from the solution and from the headspace had disappeared the reaction slowed down (Figures 12a and 13a). Copper concentrations as low as 0.01 ppm exerted a marked influence on the rate of autoxidation. At this temperature, the relation between hydrazine degraded and ammonia produced was not lineal. The ratio of ammonia produced to hydrazine degraded increased with temperature (Figure 12b and 13b).

When lower hydrazine concentrations were used (Figure 14) most of the hydrazine had degraded after one day in the 0.1 and 1.0 ppm Cu

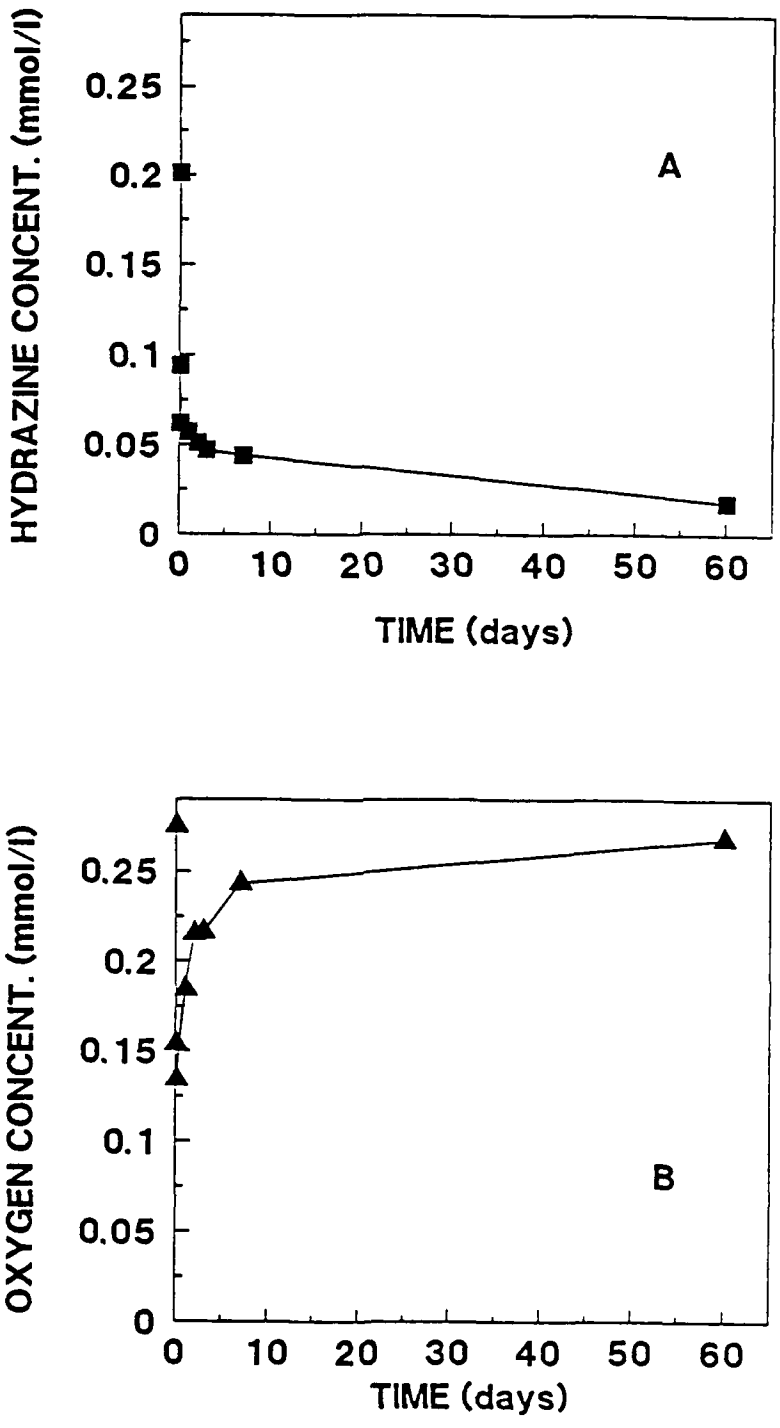


Figure 11. (A) Hydrazine degradation in 1 ppm Cu^{+2} Solution at 25°C in Serum Vials. (B) Oxygen Concentration Changes During Hydrazine Degradation at 25°C.

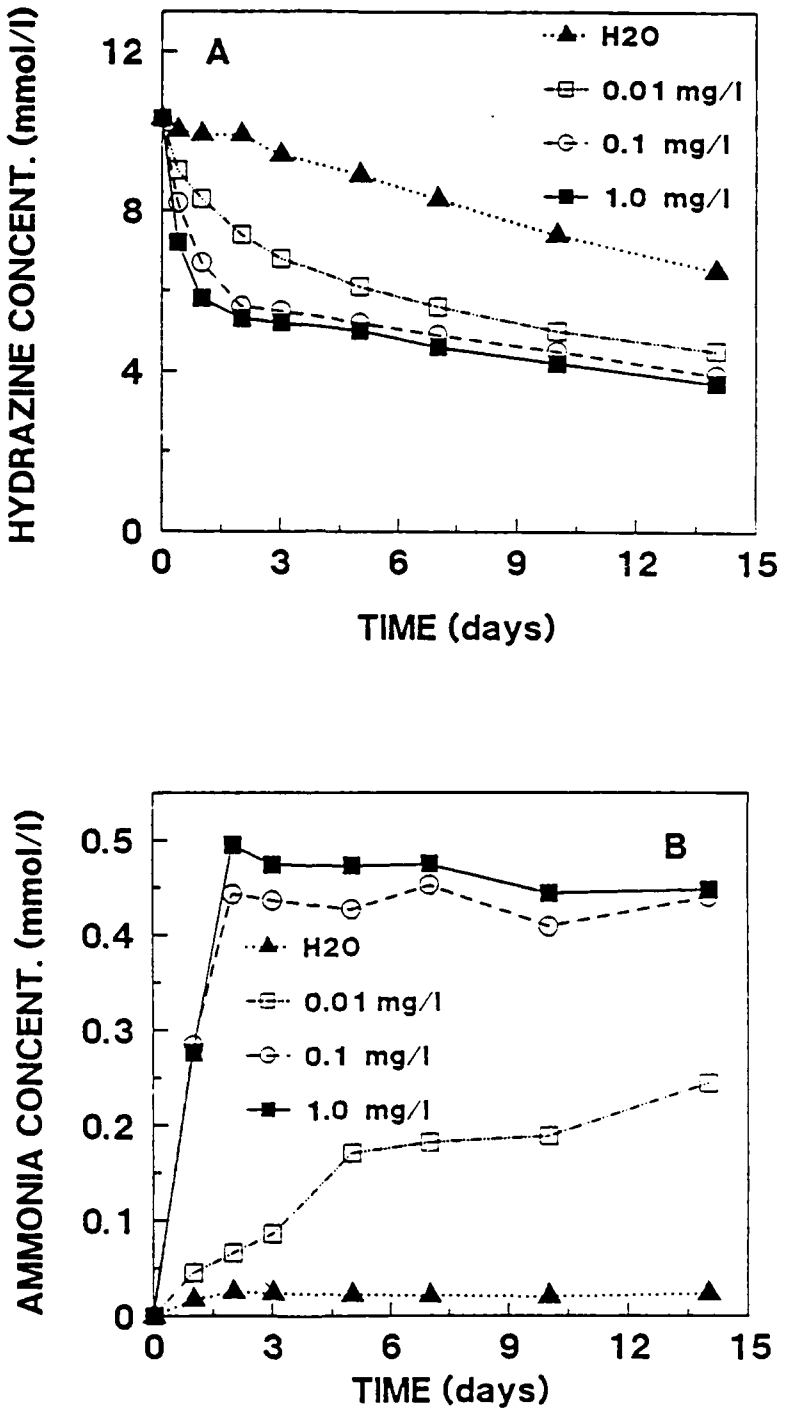


Figure 12. Hydrazine Degradation at 30°C in Serum Vials: (A) Effect of Cu^{+2} Concentration on Hydrazine Degradation, (B) Ammonia Evolution From Hydrazine Degradation at 30°C.

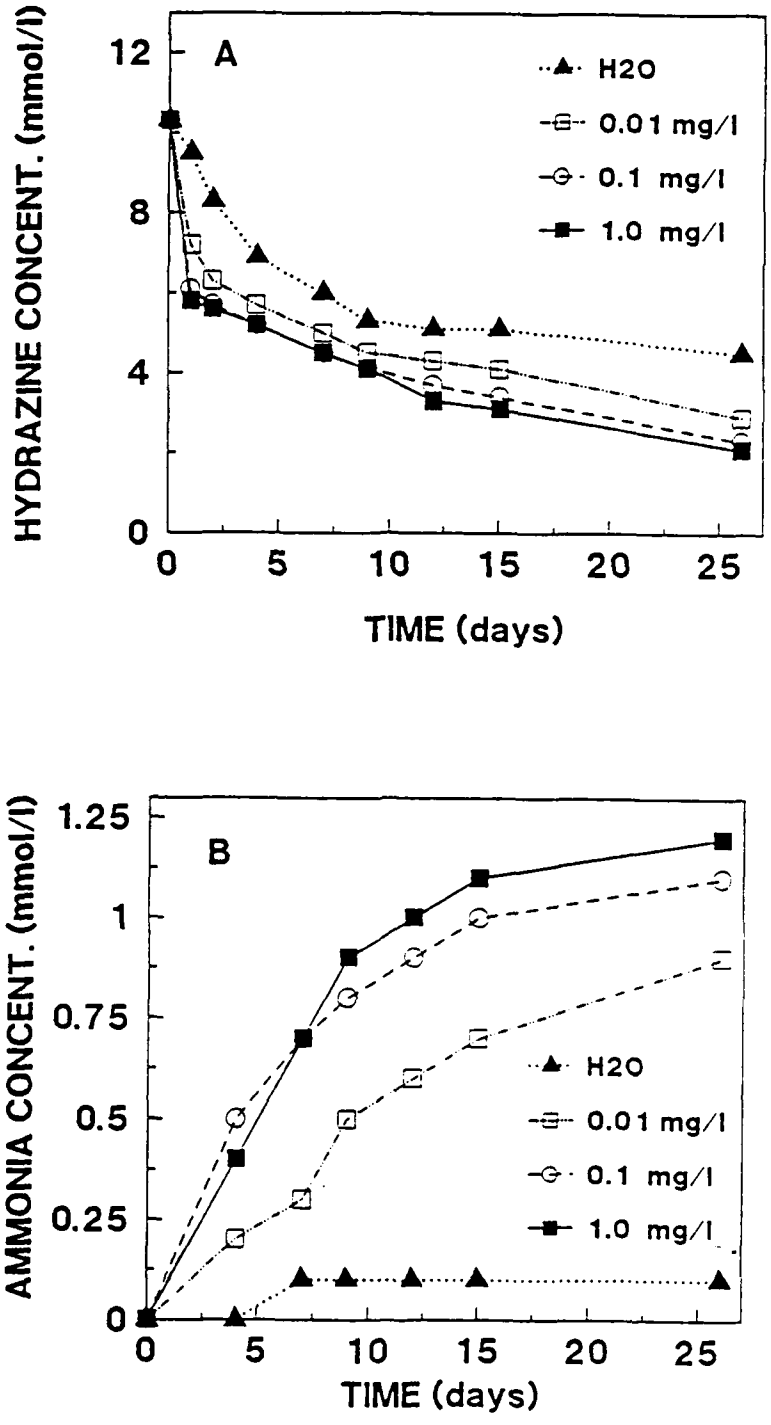
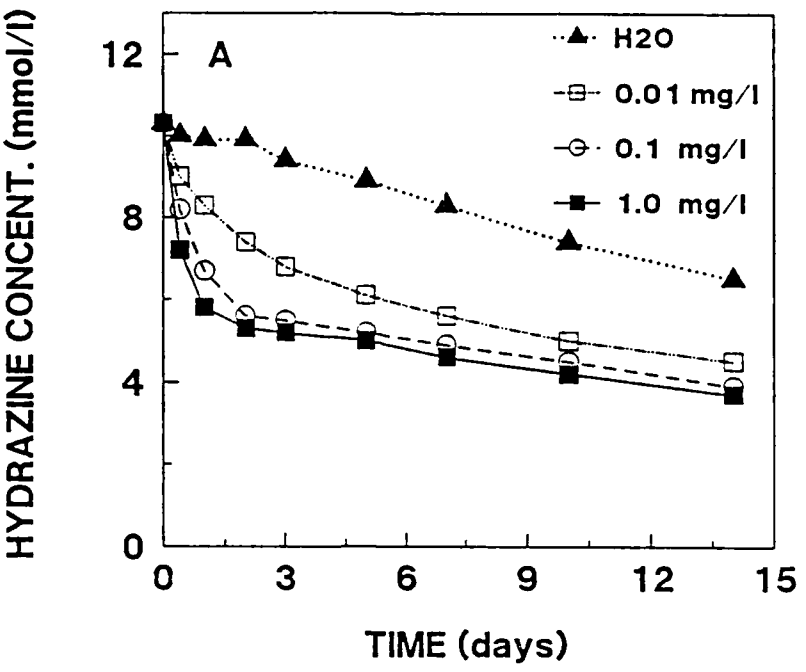


Figure 13. Hydrazine Degradation at 45°C in Serum Vials: (A) Effect of Cu^{+2} Concentration on Hydrazine Degradation, (B) Ammonia Evolution From Hydrazine Degradation at 45°C.



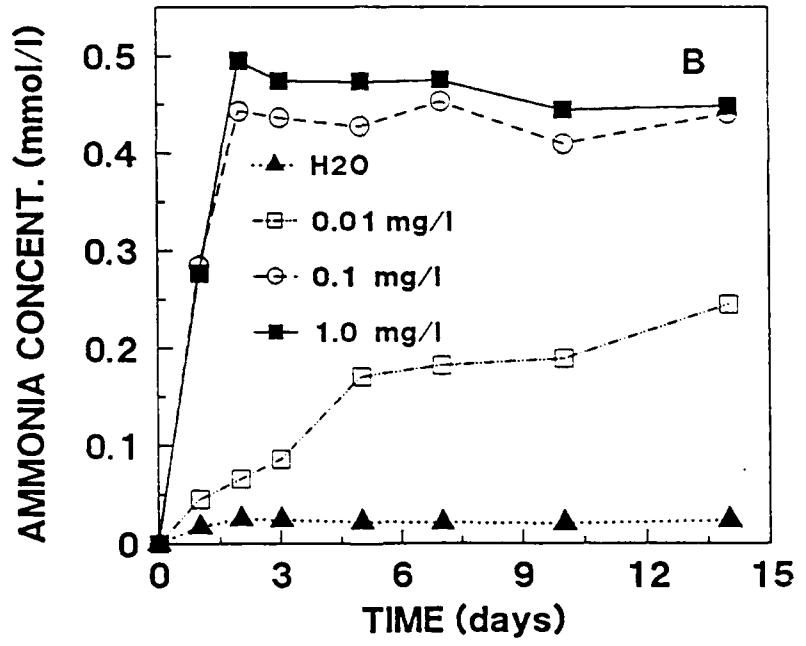


Figure 14. Hydrazine Degradation in Serum Vials at 45°C; Effect of Cu Concentration: (A) Hydrazine Degradation, (B) Ammonia Evolution from Hydrazine Degradation.

treatment. Degradation in water was slower than at high hydrazine concentration.

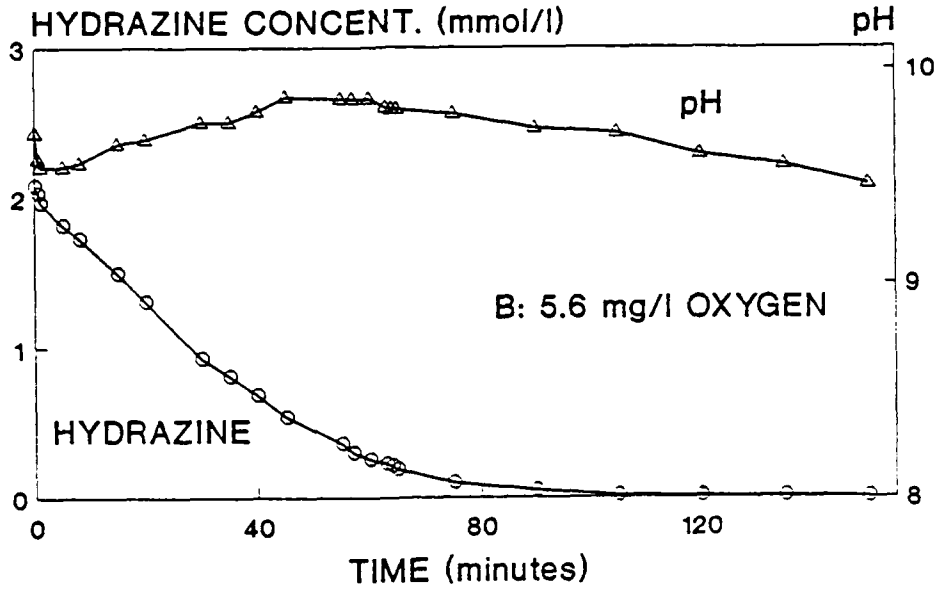
Experiments carried out in a 1.5-liters Pyrex® cell allowed control of the gas atmosphere and monitoring of changes in pH throughout the degradation. However, the trap used to remove CO_2 from the air had a slight effect on the pH of the solution. When purified nitrogen was bubbled through the cell, no degradation had taken place after 20 days.

Degradation was fast when air was bubbled through a solution containing 1 ppm Cu (Figure 15). Oxygen was consumed faster than it was supplied. Changing air to a mixture of nitrogen and 12 percent oxygen did not have a significant effect on the rate of hydrazine autoxidation. At lower Cu concentrations, the rate of degradation was affected by the pH of the solution, the rate being higher at a higher pH. When the trap used was NaOH followed by direrite there was a delay period of 6 days after which degradation took place very fast (data not shown).

a. Hydrazine Degradation in the Presence of Montmorillonite (SAz-1).

Because of an increase in the autoxidation rate of hydrazine in Cu^{+2} related to the surface of the container in contact with the solution, the reaction was probably heterogeneous. Therefore, hydrazine would probably degrade faster in a suspension that had certain amounts of Cu^{+2} distributed between the surface and the supernatant than in a solution with the same Cu^{+2} concentration but no clay. This study used a Na-montmorillonite suspension with particles smaller than 0.5 μ/m. This clay had a cation exchange capacity of 81 meq/100 g. Upon addition of an aliquot of Cu solution about 85% of the Cu^{+2} was adsorbed by the clay and 15% remained in the supernatant (Table 3).

The presence of the clay had strong effect on hydrazine degradation even when Cu was not present (Figure 16a). Addition of Cu^{+2} had a stronger effect on the solution than it did in the suspension (Figures 16b,c,d). Since the total amount of Cu was the same this shows that free Cu is a more effective catalyst than exchangeable Cu.



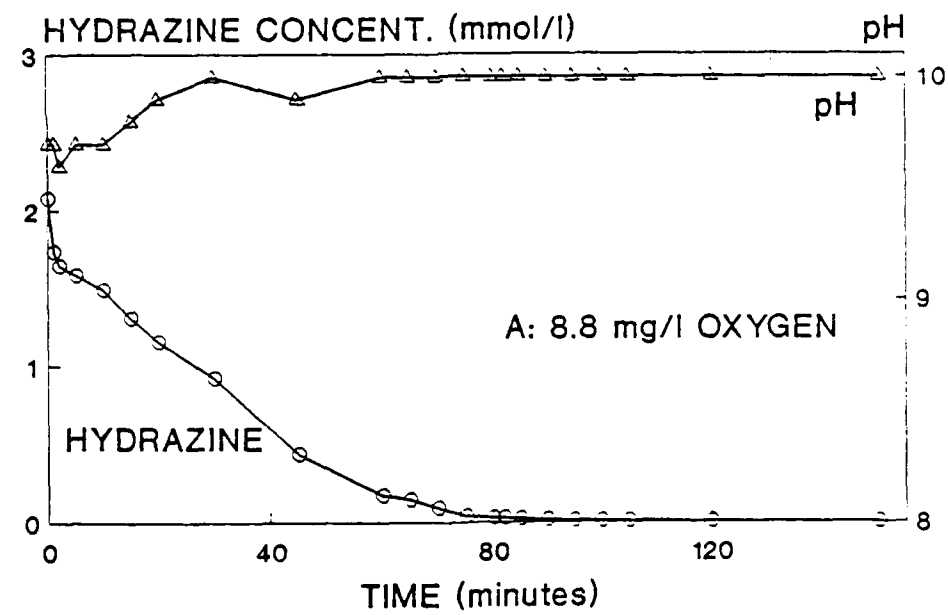


Figure 15. Hydrazine Degradation at 22°C in a 1.5 l Pyrex Cell; Effect of Cu^{+2} and Oxygen Concentration.

TABLE 3. DISTRIBUTION OF ADDED Cu^{+2} BETWEEN THE SUPERNATANT AND THE SURFACE OF THE CLAY

Clay mg/ml	Cu^{+2} added ppm	Cu^{+2} supernat. ppm	Cu^{+2} adsorb. µeq/g
2.7	0	0	0
2.7	0.01	b.d.l.*	b.d.l.
2.7	0.1	0.0145	0.98
2.7	1.0	0.19	9.4
2.7	10.0	1.31	100

* b.d.l. below detection limits

4. Adsorption Isotherms

An assumption inherent in determining adsorption isotherms is that loss of the sorbate is a measure of adsorption. The assumption is valid for most sorbates, but not so for the highly reactive hydrazine species. Earlier studies indicated that in aqueous systems with O_2 present, hydrazine undergoes an autoxidation reaction as shown below:

$$N_2H_4 + O_2 \text{ --- } N_2 + 2\ H_2O \qquad (1)$$

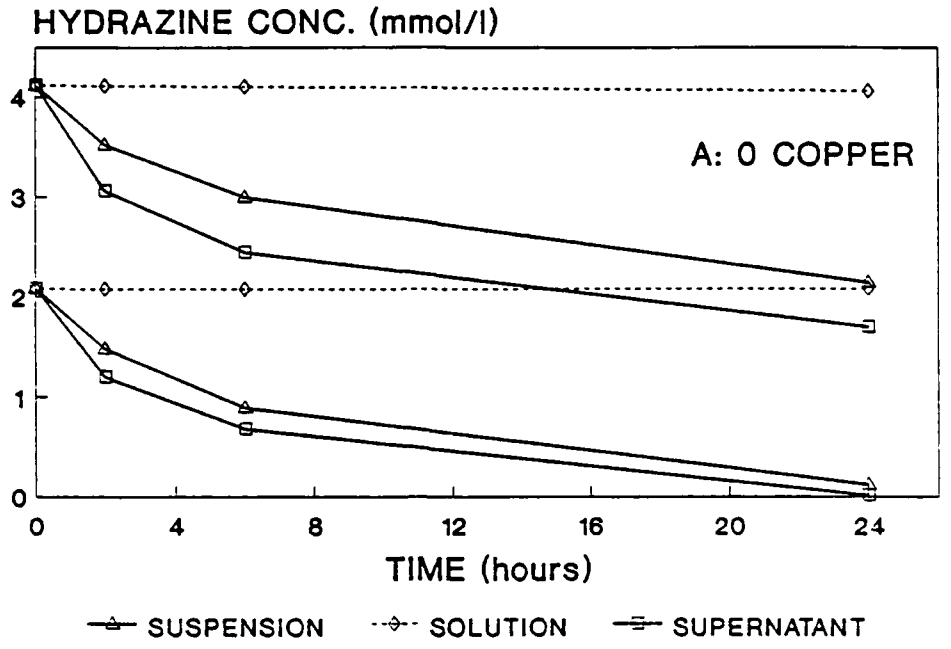
Other parameters that influence this oxidation reaction are pH, catalysts (Cu, Mn), temperature, and possibly light. Should the proper conditions exist the following reaction might also be a possible pathway for N_2H_4 loss:

$$3\ N_2H_4 \text{ --- } 4\ NH_3 + N_2 \qquad (2)$$

Exclusive of Reactions (1) and (2) the remaining possibilities for hydrazine disappearance would be as follows:

$$X\text{-}Na^+ + N_2H_5^+ \text{ --- } X\text{-}N_2H_5^+ + Na^+ \qquad (3)$$

where X- is the ion exchange sites on clays or soils and Na^+ is the counter ion. This exchange reaction is reversible and nondestructive. The other type of reaction is a sorptive reaction that may or may not be reversible:



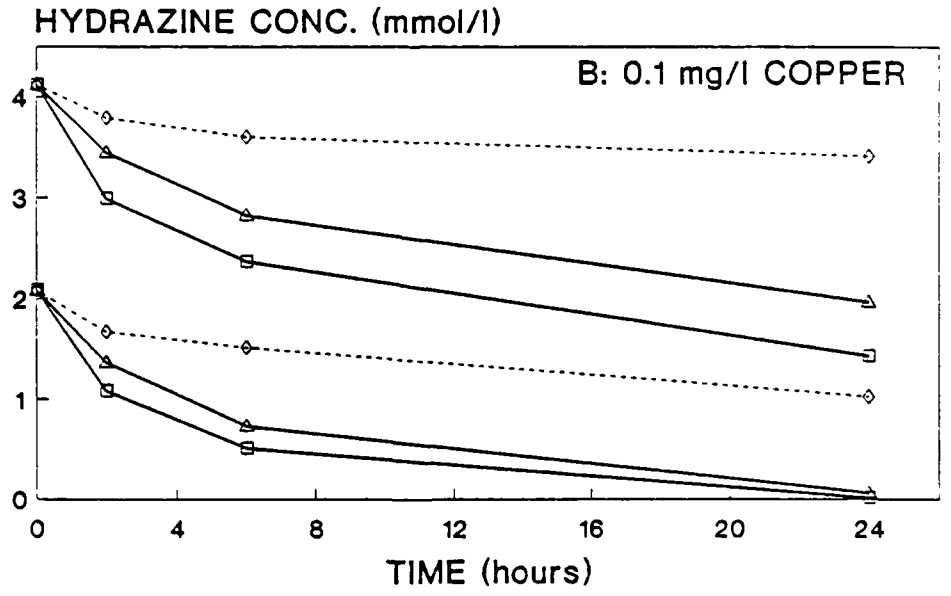
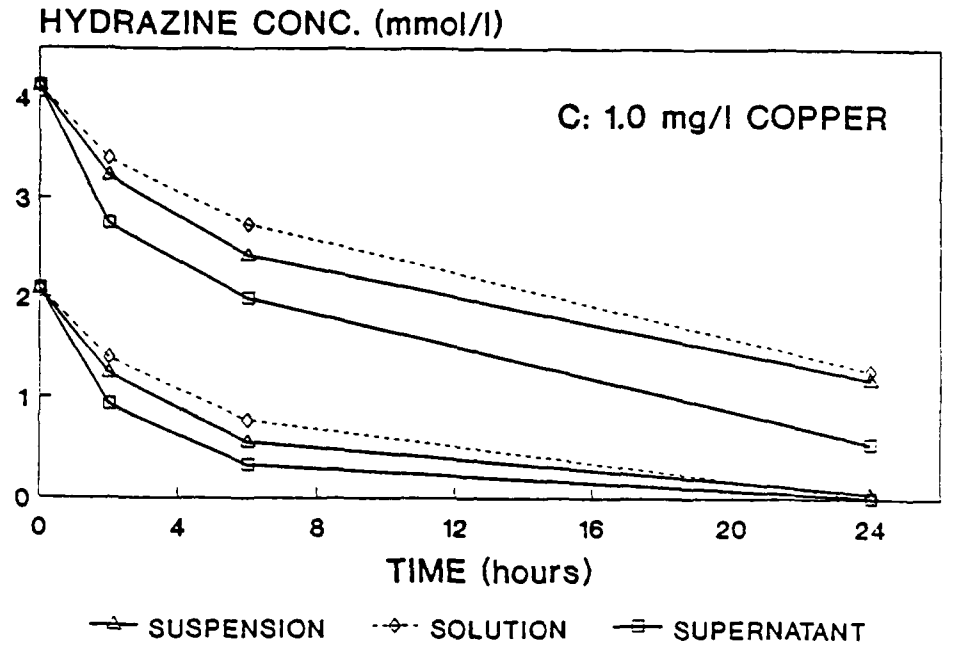


Figure 16. Effect of Cu^{+2} on Hydrazine Degradation in Solution and in the Presence of a Na-Montmorillonite (Saz-1) Suspension; (A) Absence of Cu, (B) 0.1 ppm Cu.



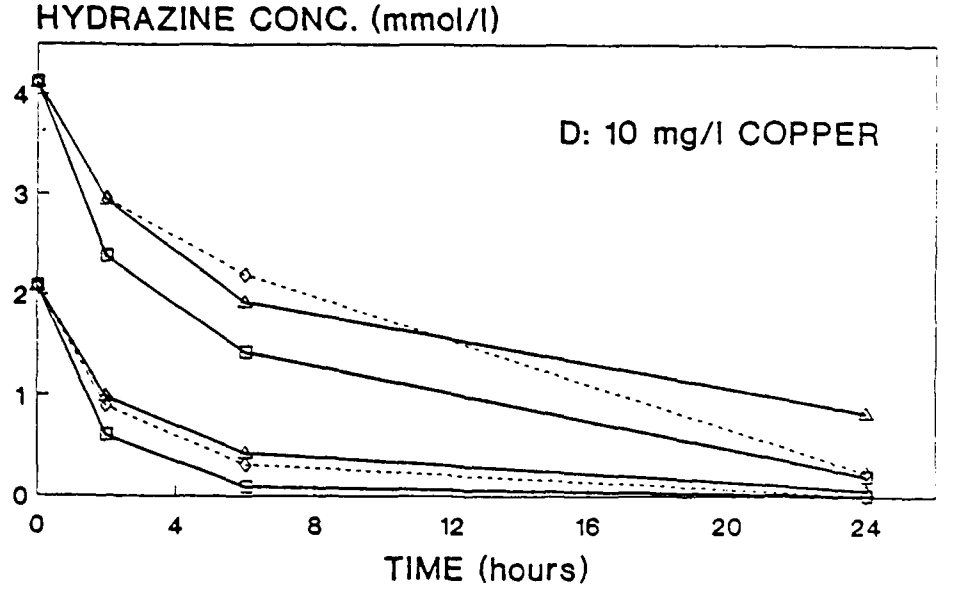


Figure 16. Effect of Cu^{+2} on Hydrazine Degradation in Solution and in the Presence of a Na-Montmorillonite (Saz-1) Suspension; (C) 1.0 ppm Cu, (D) 10.0 ppm Cu. (Concluded)

$$X-) + N_2H_4 \text{ ---- } X-)-N_2H_4 \qquad (4)$$

where X-) is the mineral or nonmineral surface that retains the hydrazine molecule.

The adsorption isotherm experiments were conducted under conditions that minimize Reactions (1) and (2) and partitions the retention of hydrazine into Reactions (3) and (4). Subsequent desorption experiments shed some light on the magnitude of irreversible adsorption of hydrazine in the system studied.

a. Adsorption of Hydrazine on Na-Kaolinite

Under acidic conditions (pH=4.0) and low hydrazine concentrations, the primary mechanism of retention was on the exchange sites. This was evident because of the concurrent appearance and disappearance of equal amounts of Na^+ and hydrazine, respectively, in the supernatant. A slight rise in the pH of the clay suspension confirms that $N_2H_5^+$ was the main species disappearing from solution because unprotonated hydrazine would have to react with protons to maintain the appropriate pKa equilibrium. At higher concentrations some hydrazine was adsorbed onto sites that had not been occupied by Na^+. No iron or silicon was detected in the supernatant before or after the addition of hydrazine (Tables 4 and 5). This would indicate that kaolinite was stable in the presence of hydrazine and that no iron coatings were reducted by hydrazine. Exhaustive washing of the clay samples with 0.1N KCl resulted in a 90 percent recovery of initially adsorbed hydrazine from the highest hydrazine additions (Figure 17).

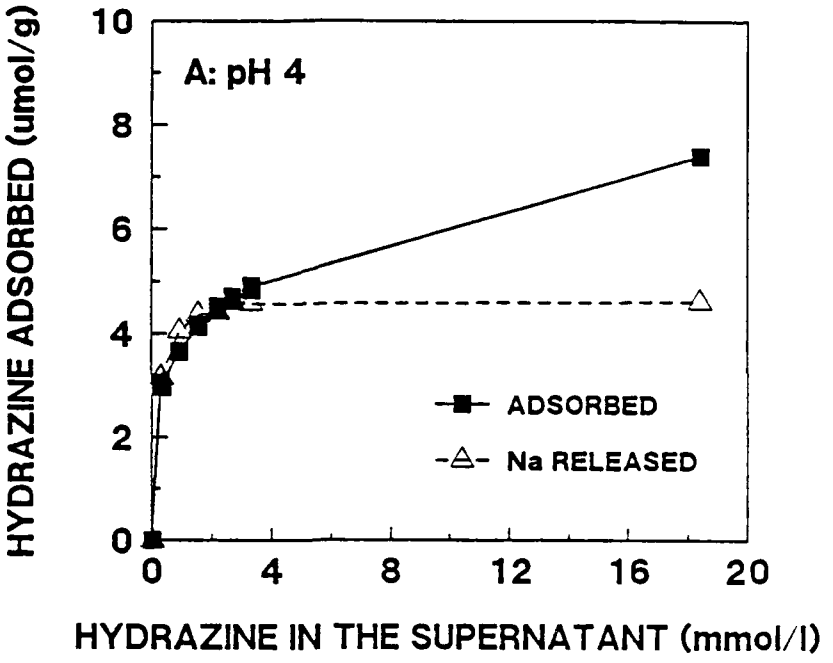
Under alkaline conditions (pH = 8.0) the pattern was different. The amount of hydrazine adsorbed at any given solution concentration was much higher than at pH 4. However, the amount of sodium displaced by hydrazine was similar to that at pH 4. Most of the adsorbed hydrazine was in the unprotonated form. This resulted in a lower clay suspension pH upon addition of hydrazine. Only 80 percent of the hydrazine adsorbed was recovered after washing with KCl (Figure 17).

TABLE 4. ANALYSIS OF KAOLINITE SUPERNATANTS AT pH 4.

Before adding hydrazine		After adding hydrazine			
Fe mmol/l	Si mmol/l	Hydrazine mmol/l	pH	Fe mmol/l	Si mmol/l
0.00	0.0	0.30	5.6	0.00	0.0
0.00	0.0	0.91	5.2	0.00	0.0
0.00	0.0	1.54	4.9	0.00	0.0
0.00	0.0	2.19	4.8	0.00	0.0
0.00	0.0	2.68	4.7	0.00	0.0
0.00	0.0	3.31	4.8	0.00	0.0

TABLE 5. ANALYSIS OF KAOLINITE SUPERNATANTS AT pH 8.

Before adding hydrazine		After adding hydrazine			
Fe mmol/l	Si mmol/l	Hydrazine mmol/l	pH	Fe mmol/l	Si mmol/l
0.00	0.0	0.12	6.8	0.00	0.0
0.00	0.0	0.45	6.4	0.00	0.0
0.00	0.0	0.92	6.4	0.00	0.0
0.00	0.0	1.41	6.8	0.00	0.0
0.00	0.0	2.20	7.4	0.00	0.0
0.00	0.0	3.04	7.5	0.00	0.0



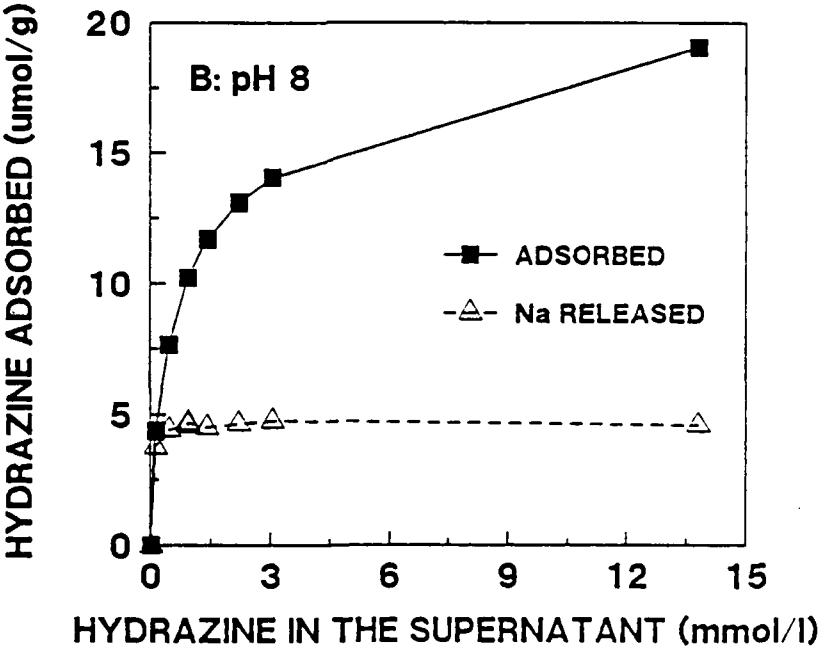


Figure 17. Adsorption Isotherms of Hydrazine on Na-Kaolinite (KGa-1): (A) pH 4, (B) pH 8.

b. Adsorption of Hydrazine on Na-Montorillonite

The initial adsorption isotherm on Na-montmorillonite was done without controlling the pH (Figure 18). The fact that adsorption decreased at high hydrazine concentrations revealed that hydrazinium produced a pH increase decreasing the concentration of hydrazinium in solution even more.

The following set of isotherms under controlled pH confirmed that the main mechanism of adsorption in this clay suspension was cation exchange, both at pH 4 and 8. In both cases, the pH of the suspension increased up to 2 units. At pH 8 the amount of Na released by the clay was higher than the amount of hydrazine being adsorbed. After analyzing for silicon and iron we found that small amounts of montmorillonite had dissolved (Tables 6 and 7).

TABLE 6. ANALYSIS OF MONTMORILLONITE SUPERNATANTS AT pH 4.

Before adding hydrazine		After adding hydrazine			
Fe mmol/l	Si mmol/l	Hydrazine mmol/l	pH	Fe mmol/l	Si mmol/l
0.03	6.3	0.01	6.6	0.06	16.8
0.03	6.3	0.02	6.4	0.06	11.9
0.03	8.3	0.07	6.2	0.04	6.0
0.03	8.3	0.18	6.2	0.00	0.9
0.03	4.3	0.33	6.0	0.00	0.4
0.03	3.6	0.53	5.8	0.00	0.4

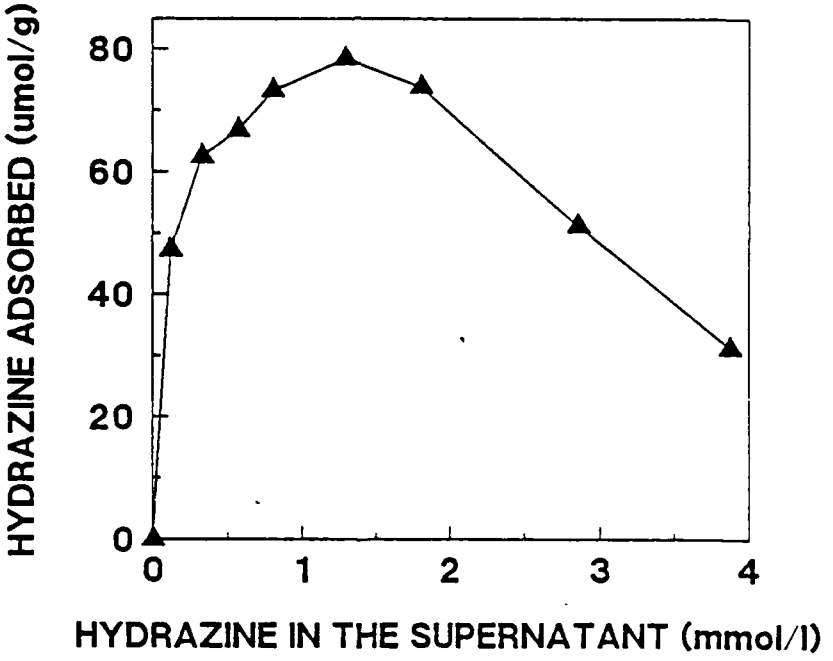


Figure 18. Adsorption Isotherm of Hydrazine on Na-Montmorillonite (SAz-1) Without Controlling the pH of the Suspension.

TABLE 7. ANALYSIS OF MONTMORILLONITE SUPERNATANTS AT pH 8.

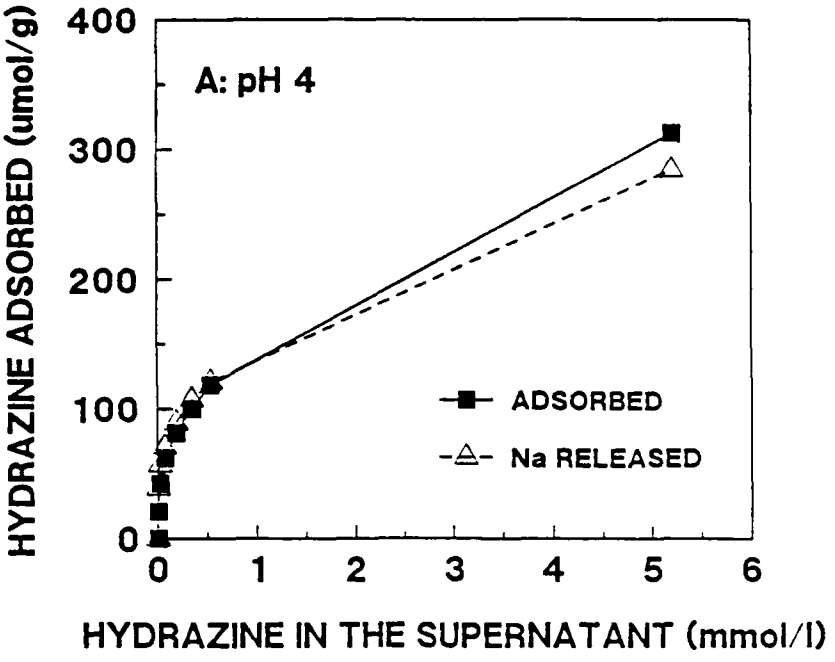
Before adding hydrazine		After adding hydrazine			
Fe mmol/l	Si mmol/l	Hydrazine mmol/l	pH	Fe mmol/l	Si mmol/l
0.03	9.4	0.13	9.5	0.04	8.8
0.07	13.2	0.49	9.8	0.05	11.2
0.04	4.2	0.98	9.4	0.04	8.1
0.04	3.7	1.57	9.4	0.03	5.7
0.04	6.1	1.72	9.5	0.06	16.1
0.03	3.6	2.29	9.2	0.05	10.3

The excess sodium in the supernatant was probably associated with the dissolved clay. Adsorption of hydrazine was lower in the montmorillonite suspension at pH 8. This rationale is logical because the pKa of hydrazine indicates that the majority of the hydrazine at this pH is in the neutral form rather than the $N_2H_5^+$ form. Washing the clay with KCl desorbed 60% of the adsorbed hydrazine at pH 4.0 and the same percentage was recovered at pH 8.0 (Figure 19).

6. Adsorption of Hydrazine on Arredondo Soil Horizons

Selected chemical properties of Arredondo soil top horizons are presented in Table 8. The most important difference among them with respect to hydrazine adsorption are a higher clay and organic matter content in the Ap horizon and a lower concentration of metals in the E_2 horizon.

The first set of isotherms from Arredondo horizons is presented in Figures 20, 21, and 22. The isotherms were carried out in an anaerobic incubator, maintaining a constant ionic strength. Hydrazine adsorption was correlated to organic matter and clay content. Adsorption



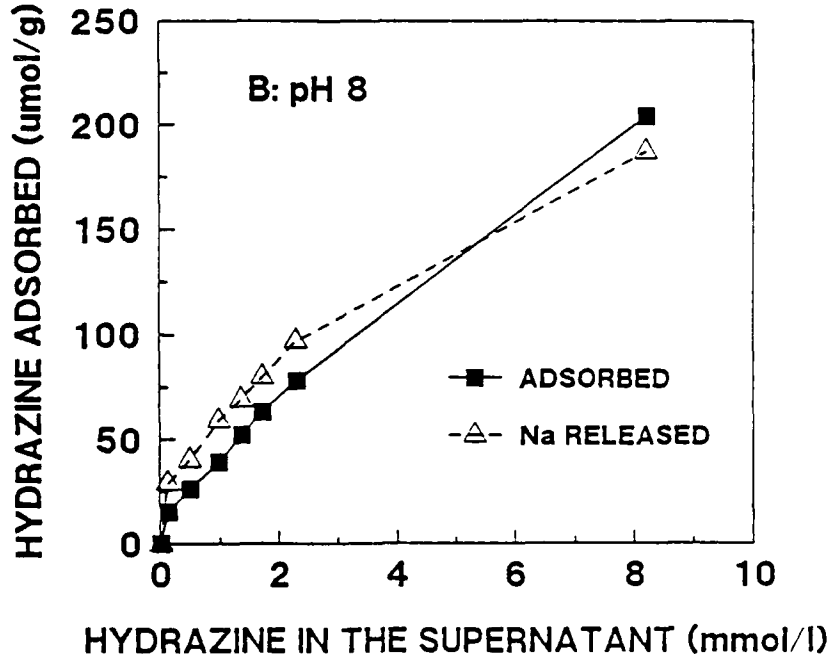
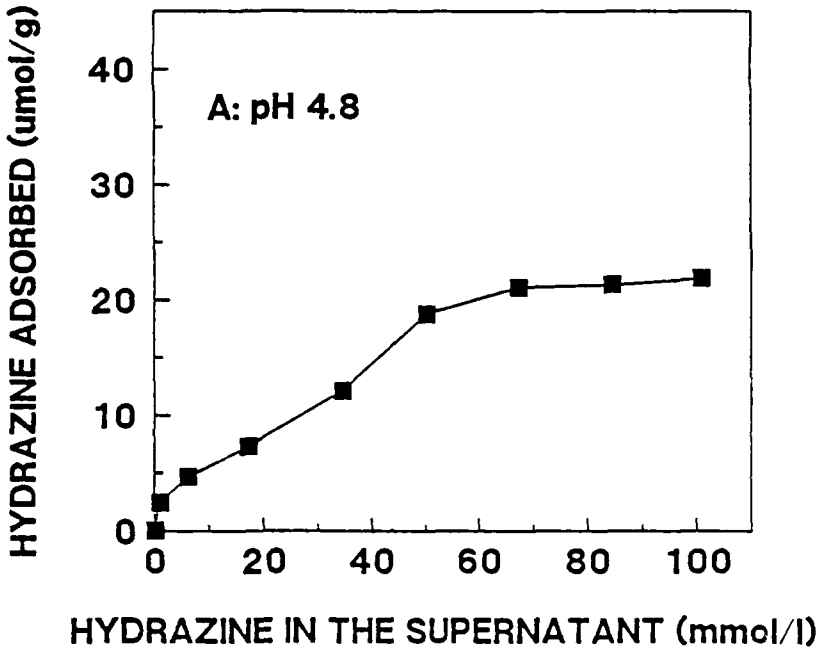


Figure 19. Adsorption Isotherms of Hydrazine on Na-Montmorillonite: (A) pH 4, (B) pH 8.



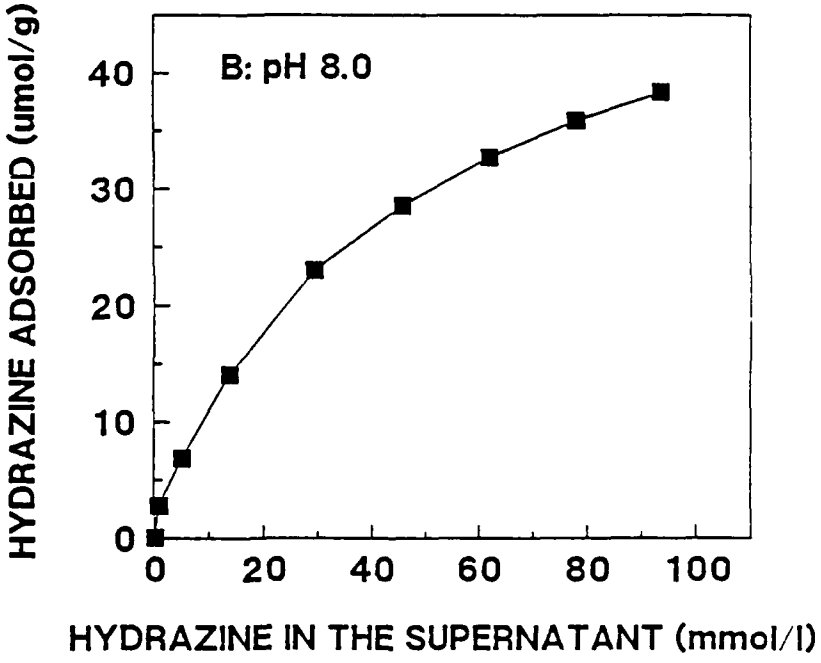
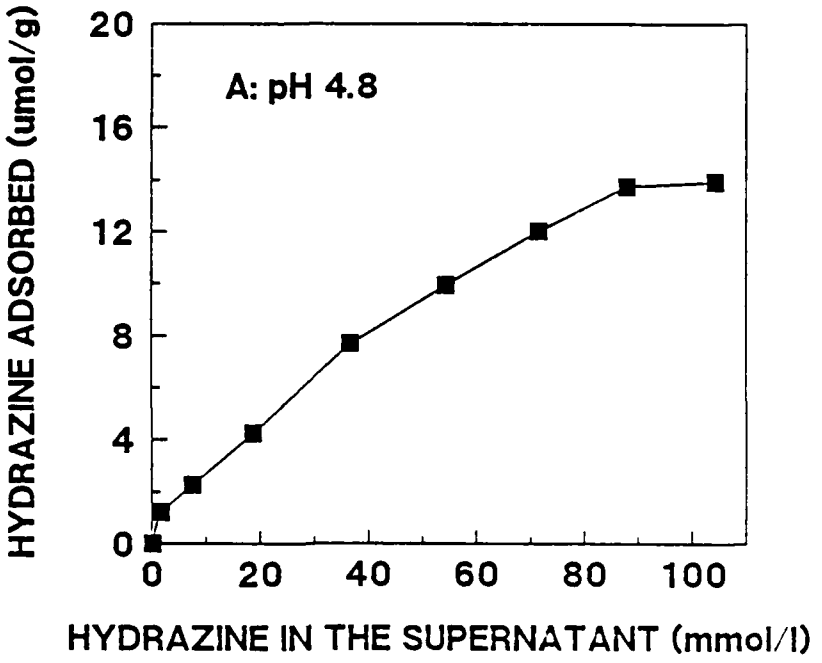


Figure 20. Adsorption Isotherms of Hydrazine on Arredondo-Ap (Ionic Strength 0.01 N).



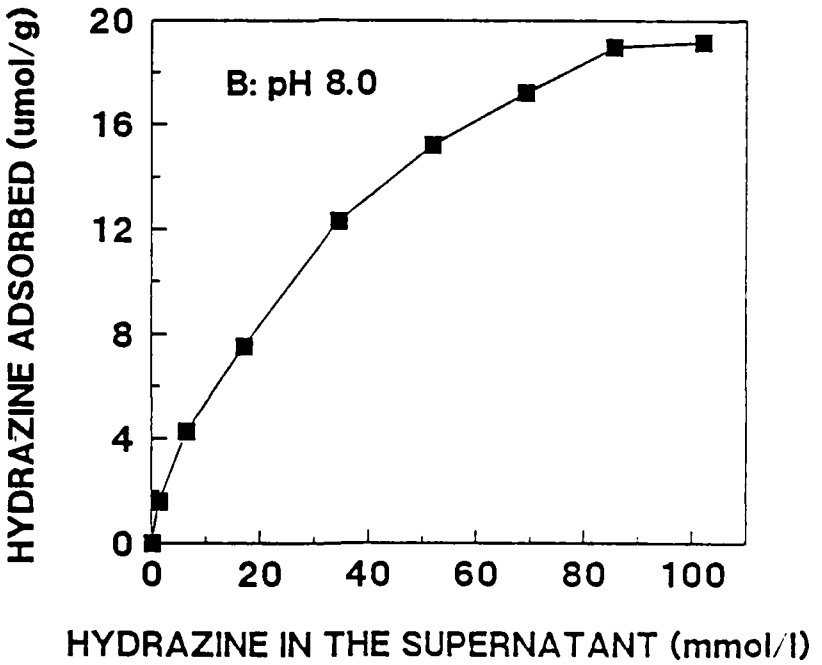
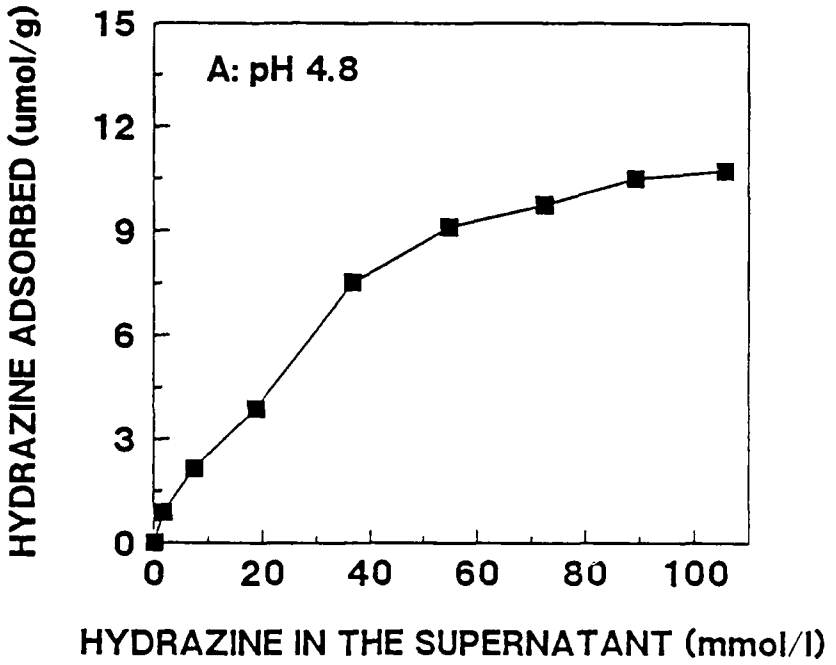


Figure 21. Adsorption Isotherms of Hydrazine on Arredondo-El (Ionic Strength 0.01 N)



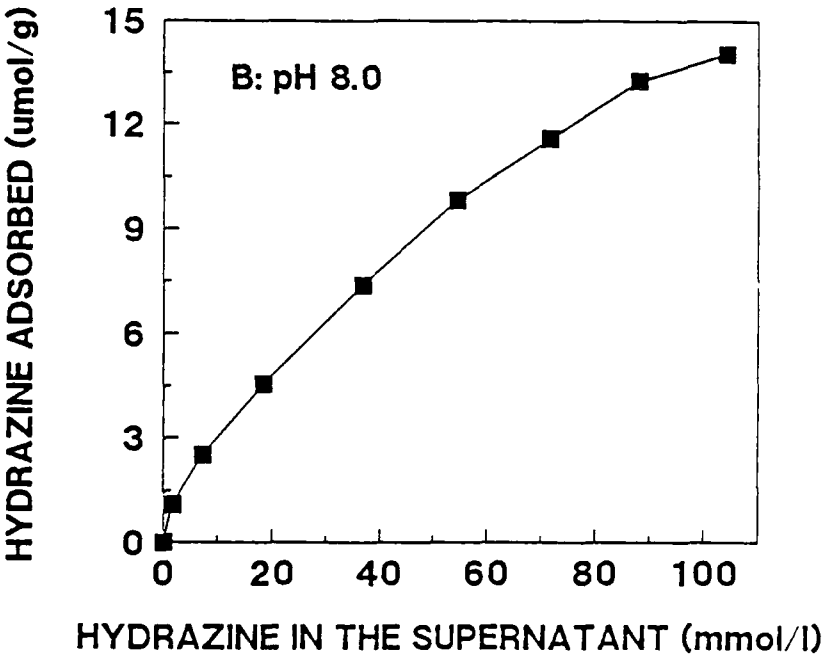


Figure 22. Adsorption Isotherms of Hydrazine on Arredondo-E2 (Ionic Strength 0.01 N).

TABLE 8. PROPERTIES OF ARREDONDON SOIL TOP HORIZONS

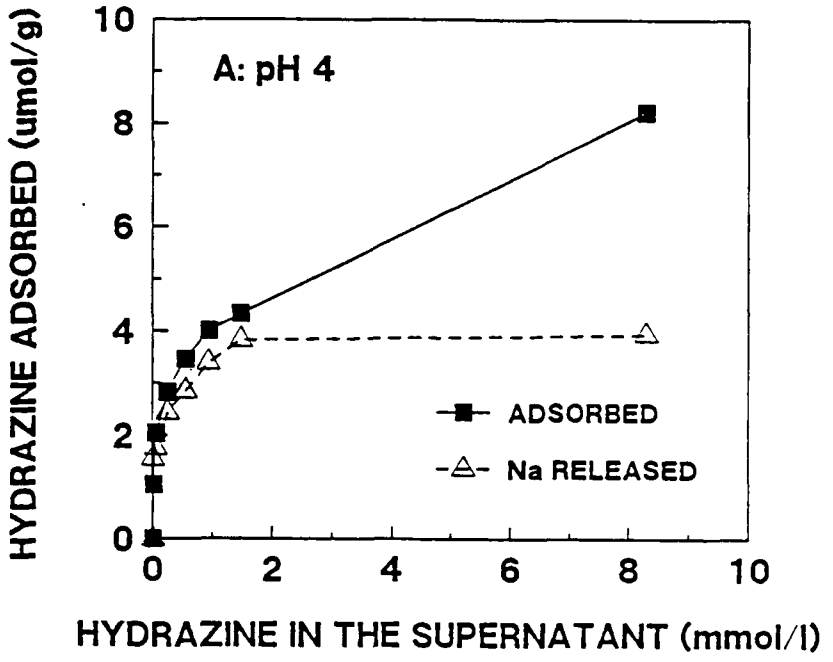
Horizon	% Clay	% O.M.	pH	Cu	Mn	Al	Fe
					MG / Kg soil		
Ap	2.6	1.84	6.0	0.04	9.6	221	17.6
E_1	1.7	0.34	5.9	0.24	2.5	250	17.6
E_2	1.8	0.14	5.4	0.16	1.2	86	10.4

was higher at pH 8 in all three horizons indicating that both species, neutral hydrazine and hydrazinium, were adsorbed. Because ionic strength was maintained with $CaCl_2$, hydrazinium ion had to compete with Ca^{+2} for the exchange sites and adsorption was probably lower.

In the second set of isotherms from Arredondo soil horizons, total hydrazine adsorbed and hydrazine in exchangeable sites were separated by measuring Na^{+1} displaced from the surface of the clay to the supernatant by the hydrazinium ion (*Figures 23 and 24*).

The main mechanism of adsorption at pH 4 and low hydrazine concentrations in the Arredondo Ap horizon was cation exchange. However, at higher concentrations more than 50 percent of the hydrazine interacted with a different kind of binding site. Because the Ap horizon had almost 2 percent organic matter it appears that hydrazine was adsorbed on organic surface functional groups such as carbonyl groups. No hydrazine was recovered during the desorption process with 0.1 N KCl. The results were very similar at pH 8.0 except for a higher adsorption at high concentrations on non exchangeable sites. At this pH, 20 percent of the hydrazine initially adsorbed was extracted.

Analysis of the supernatant for the Arredondo Ap and E2 soil horizons are shown in Tables 9, 10, 11, and 12. Under acidic conditions (pH 4.0) it appears that the main species removed from solution was the hydrazinium ion which resulted in an increase in solution pH (Table 9 and 11). Under alkaline conditions (pH 8.0), there was a subsequent decrease in pH as a result of hydrazine (the neutral species) being the main species removed from solution. These shifts in pH are a result of the pKa of hydrazine as the solution obtains equilibrium.



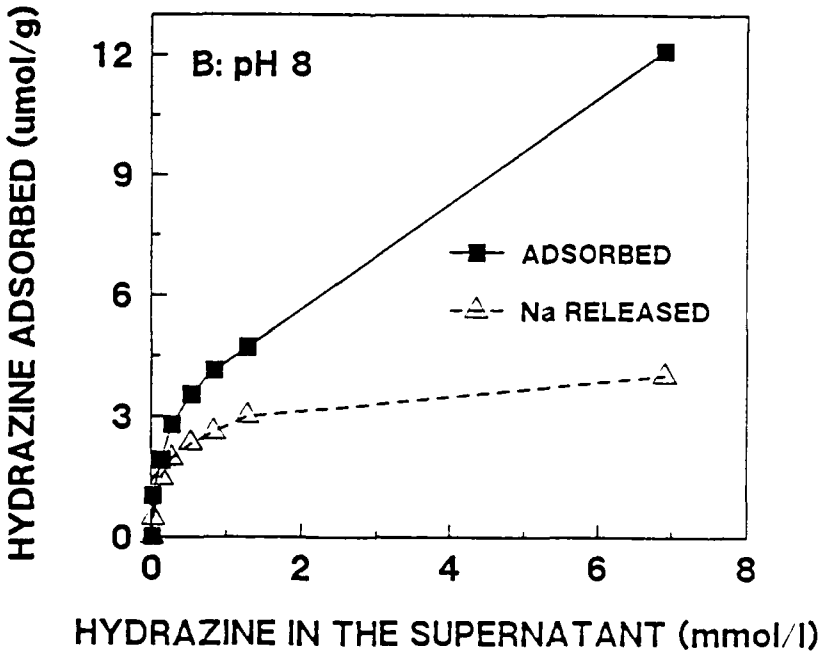


Figure 23. Adsorption Isotherms of Hydrazine on Arredondo-Ap.

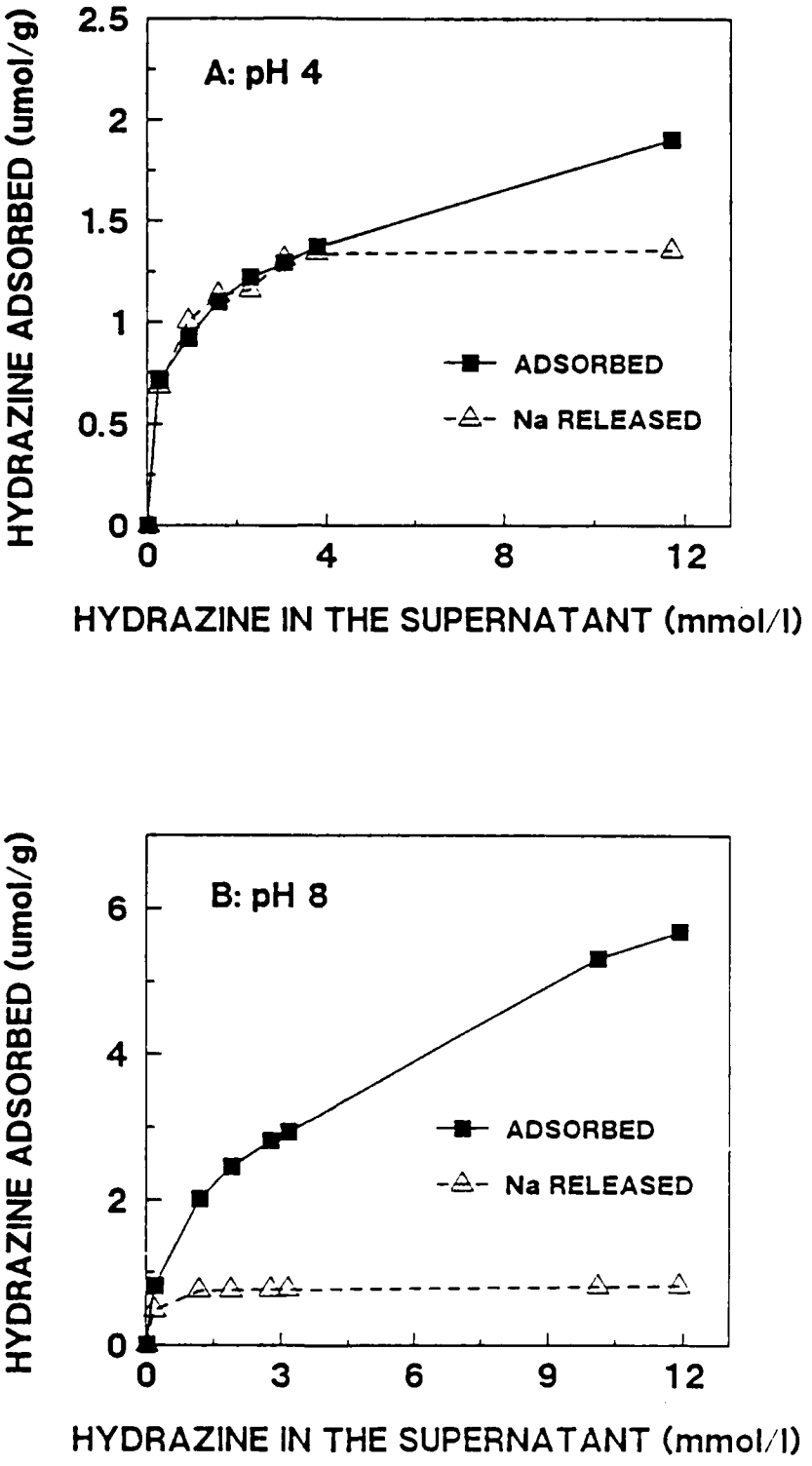


Figure 24. Adsorption Isotherms of Hydrazine on Arredondo-E2.

TABLE 9. ANALYSIS OF ARREDONDO Ap SUPERNATANTS AT pH 4.

Before adding hydrazine		After adding hydrazine			
Fe mmol/l	Si mmol/l	Hydrazine mmol/l	pH	Fe mmol/l	Si mmol/l
0.14	1.3	0.01	6.4	0.15	1.3
0.12	1.1	0.05	6.0	0.12	1.0
0.12	1.2	0.25	5.7	0.00	0.2
0.15	1.5	0.54	5.4	0.00	0.5
0.13	1.3	0.93	5.3	0.01	0.5
0.16	1.5	1.47	5.0	0.02	0.4

TABLE 10. ANALYSIS OF ARREDONDO-Ap SUPERNATANTS AT pH 8.

Before adding hydrazine		After adding hydrazine			
Fe mmol/l	Si mmol/l	Hydrazine mmol/l	pH	Fe mmol/l	Si mmol/l
0.08	0.5	0.02	6.6	0.16	1.7
0.08	0.4	0.14	6.4	0.16	1.8
0.08	0.4	0.27	6.5	0.19	1.7
0.08	0.5	0.52	6.5	0.16	1.2
0.07	0.6	0.83	6.6	0.20	1.7
0.09	0.6	1.28	6.7	0.20	1.7

TABLE 11. ANALYSIS OF ARREDONDO-E2 SUPERNATANTS AT pH 4.

Before adding hydrazine		After adding hydrazine			
Fe mmol/l	Si mmol/l	Hydrazine mmol/l	pH	Fe mmol/l	Si mmol/l
0.00	0.0	0.25	5.8	0.00	0.3
0.00	0.0	0.90	5.5	0.00	0.2
0.00	0.0	1.57	5.3	0.00	0.5
0.00	0.2	2.28	5.1	0.00	0.5
0.00	0.1	3.04	4.9	0.00	0.5
0.00	0.1	3.77	4.8	0.00	0.5

TABLE 12. ANALYSIS OF ARREDONDO-E2 SUPERNATANTS AT pH 8.

Before adding hydrazine		After adding hydrazine			
Fe mmol/l	Si mmol/l	Hydrazine mmol/l	pH	Fe mmol/l	Si mmol/l
0.04	0.8	0.18	7.0	0.16	1.5
0.07	0.8	1.18	7.0	0.01	0.4
0.08	0.8	1.87	7.0	0.01	0.5
0.07	0.8	2.76	7.0	0.01	1.2
0.05	0.5	3.17	7.3	0.01	0.6
0.04	0.4	11.90	7.7	0.01	0.6

Arredondo E_2 horizon adsorbed less hydrazine than the Ap horizon. Under acidic conditions the main mechanism of adsorption was cation exchange specially at low hydrazine concentrations. Under alkaline conditions (pH 8) hydrazine was adsorbed mainly in nonexchangeable sites. A summary of the adsorption data for all materials tested are shown in Figure 25.

D. CONCLUSIONS

Hydrazine autoxidation in solution appeared to follow the reaction in Equation (1).

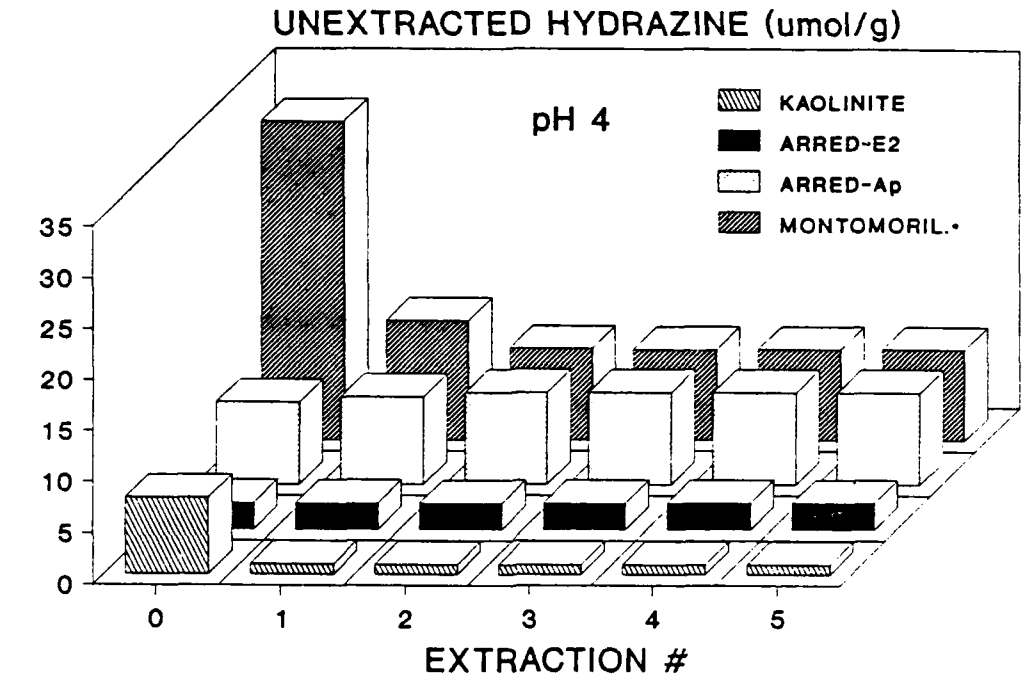
$$N_2H_4 + O_2 \text{ --- } N_2 + 2H_2O \qquad (1)$$

Hydrazine autoxidation occurred only at pH values above 4.0. Autoxidation did not occur when the pH was less than 4.0 even in the presence of a catalyst. At pH values above 4.0 the rate of reaction was first-order with respect to Cu concentration. Other factors observed to affect the rate of hydrazine autoxidation in solution were buffer concentrations (phosphate), ionic strength, and temperature. For the two natural waters studied a half-life of 8 to 12 days was calculated.

Hydrazine degraded faster in the presence of the clay minerals kaolinite and montmorillonite than in clay-free solutions. When Cu^{2+} was present in the clay studies the rate of hydrazine degradation was enhanced above that of clay alone. This was a result of the free Cu^{2+} in solution rather than a clay-Cu^{2+} surface reaction.

The nature and extent of hydrazine adsorption by clays and soils is highly dependent on the types of surface functional groups present in the solid surfaces. Under acidic conditions (pH 4.0) 99.9% of the hydrazine occurred as the protonated species ($N_2H_5^+$) and should have been able to readily replace Na^+ from exchange sites. Under alkaline conditions (pH 8.0), 50% of the hydrazine was protonated and 50% was in neutral form.

In the case of kaolinite, where most of the surface functional groups consists of the inorganiuc OH groups exposed on broken edges,



• CONCENTRATION / 10

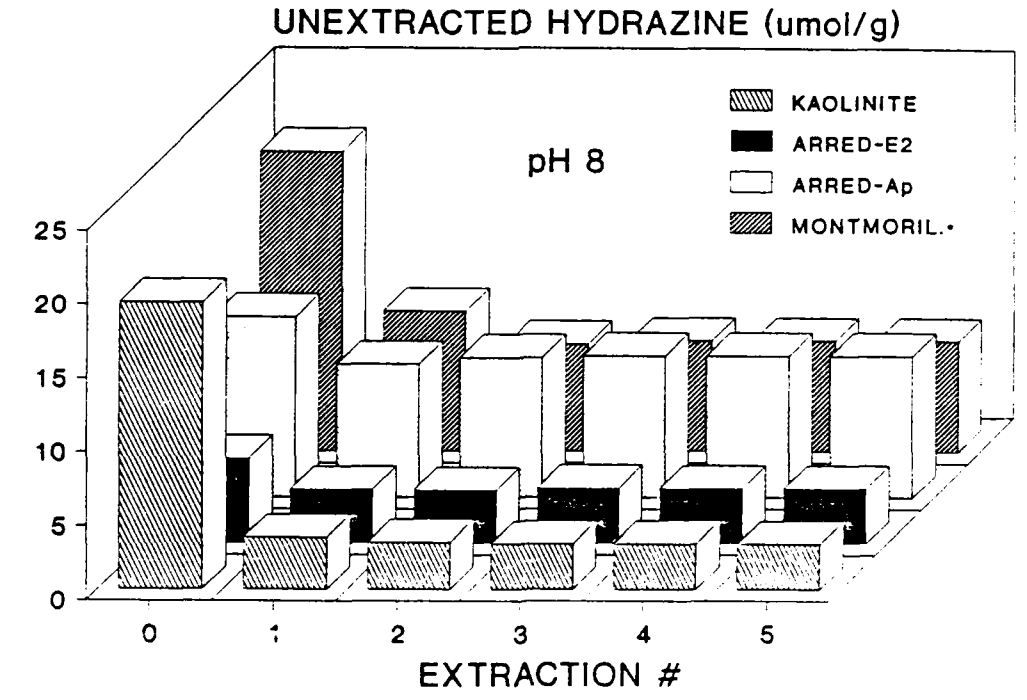


Figure 25. Hydrazine Extraction From Kaolinite and Montmorillonite Clays and From Arredondo Ap and E2 Horizons at pH 4 and pH 8.

hydrazine primarily replaced Na^+ from exchange sites under acidic conditions. Under alkaline conditions, the same amount of Na^+ was displaced by hydrazine as at pH 4.0; however, most of the adsorbed hydrazine was retained on sites that had not previously been occupied by Na^+. The most likely mechanism would be H bonding to the siloxane ditrigonal cavity on the outer surface of the clay particles. Ninety percent of the hydrazine adsorbed at pH 4 was removed by KCl washing, whereas 80% was removed at pH 8.0.

Montmorillonite has a much larger CEC due to isomorphic substitution in its octahedral layer. It primarily retained hydrazinium ion on exchange sites, both under acidic and alkaline conditions. Hydrazine adsorption was lower at pH 8.0 due to the reduced amounts of hydrazine ion under alkaline conditions. Adsorption of small amounts of $N_2H_5^+$ at pH values near the pKa resulted in equilibrium shift favoring the N_2H_4 species, an increase in pH, and a subsequent reduction in the $N_2H_5^+$ available for exchange. Only 60% of the hydrazine adsorbed at both pH values could be removed with KCl. The remaining 40% was either irreversibly adsorbed or could have been degraded. A possible mechanism for irreversible adsorption could be through complexation with iron in the structure of the clay.

Even though the clay content (primarily kaolinite) in the upper horizons of the Arredondo soil was higher than the organic matter content, hydrazine adsorption was better correlated to organic matter content. This reflects its higher CEC and affinity of the organic reactive groups for hydrazine. Under acidic conditions and at low hydrazine concentrations, hydrazine primarily replaced Na^+ from exchange sites; however, none of the hydrazine adsorbed at pH 4.0 was removed from any of the two horizons by repeated washings with KCl, and only a small fraction was extracted by 0.1 N HCl. We believe that, after hydrazine was retained on the exchange sites of organic matter, hydrazine reacted further with other neighboring groups in the organic structure, thus preventing its extraction with KCl. Under alkaline conditions (pH 8) hydrazine was adosrbed more readily than at pH 4; however, the amounts of sodium released were slightly smaller. This indicates a preference for N_2H_4 at this pH. Most of the hydrazine that had replaced Na^+ was removed by a single washing with KCl.

SECTION III

SURFACE CHEMISTRY OF HYDRAZINE

A. INTRODUCTION

The attenuation of hydrazine and its methyl derivatives by soil and aquifer constituents is of interest to the United States Air Force because of their use in several Air Force weapon systems and their adverse biological activity to man. Although the macroscopic adsorption behavior of hydrazine by clay minerals has been studied (Reference 1), little is known about the chemical mechanism(s) of interaction. Adsorption date are macroscopic intrinsically insensitive to molecular phenomena; adsorption data cannot be interpreted to obtain unequivocal, molecular-level information about the adsorbed species (Reference 8). The objective of this research effort will be to examine the molecular-level interactions of hydrazine with kaolinite using several non-invasive, *in situ spectroscopic methods.*

The vibrational spectrum of kaolinite, a ubiquitous 1:1 clay mineral found throughout the world, has been studied more intensively than that of any other 1:1,2:1. or 2:2 clay mineral (Reference 9). This spectroscopic interest in kaolinite is, in part, a result of its well-resolved, sharp hydroxyl stretching bands. In contrast to the broad vibrational linewidths of hydrous metal oxides and 2:1 clay minerals that typically have bandwidths of greater than 50-100 cm^{-1}, the measured full-width-at-half-maximum (FWHM) values for the hydroxyl stretching bands of kaolinite range (Reference 10) between 5 and 12 cm^{-1}. These comparatively sharp vibrational bands of kaolinite provide surface-sensitive probes of changes in the chemical environment surrounding these hydroxyl groups.

A more complete vibrational analysis of adsorbate-surface complexes can be obtained when perturbed vibrational modes of the adsorbate and of the surface are observed. This study will employ the surface hydroxyl groups of kaolinite as molecular probes of the interation between kaolinite and hydrazine upon formation of the intercalation complex. Changes in the frequency, intensity, and lineshape of the adsorbed species can provide direct information about the structure of the adsorbed species

and what chemical functional group(s) of the adsorbate are involved in bonding to the surface. However, these data do not provide unambiguous information about the orientation of the adsorbed species, or about which surface functional groups are involved in bonding to the adsorbate. This information can best be obtained by observing the perturbed vibrational models of the substrate. The well-resolved IR- and Raman-active bands of kaolinite should allow perturbed vibrational modes of the substrate to be resolved.

B. EXPERIMENTAL

1. Clay Mineral Preparation for FT-IR Analysis

The clay mineral samples studied were obtained from the Source Clays Repository located at the University of Missouri and operated by the Clay Minerals Society. The kaolinite sample was the well-crystalline KGa-1 Georgia-kaolinite collected from Washington county, Georgia, and the montmorillonite samples was the Saz-1 Cheto-montmorillonite collected in Apache county, Arizona. A complete description of the physical properties of these clay samples has been given by van Olphen and Fripiat (Reference 11). In addition, Raman and IR spectra of the KGa-1 kaolinite clay have been reported by Johnston et. al. (Reference 10). The collodial behavior of 1:1 clay minerals (i.e., kaolinite and serpentine group minerals) are fundamentally different from that of the 2:1 clays; therefore, separate clay mineral preparation and purification procedures were used for preparing the KGa-1 and SAz-1 clay materials.

The procedure used to prepare the SAz-1 Cheto-montmorillonite clay sample was similar to that described by Sposito et al. (Reference 12). Sixty grams of the crude reference clay were placed in 1 liter of distilled, deionized water and mixed for 2 hours with a mechanical stirrer. The fraction having an equivalent-spherical-diameter (e.s.d.) of $<0.5\mu m$ in suspension was separated by centrifugation and then flocculated by adding 800 mL of a solution containing 0.001 M HCl in 1M NaCl. The flocculated clay in the NaCl-HCl solution was centrifuged for

15 minutes at 5000 rpm on a Sorvall SS-3 centrifuge equipped with a Model G.S.A. head. After the clear supernatant solution was carefully decanted, the SAz-1 clay plug at the bottom of the centrifuge tube was redispersed manually into a fresh NaCl-HCl solution and the suspension was shaken on try shaker for 20 minutes. The suspension was centrifuged again as described above. This washing procedure was repeated about three times until the pH value of the supernatant solution dropped to 3.0. After the final NaCl-HCl wash and centrifugation, the Saz-1 clay plugs were redispersed into a 0.1 M NaCl solution and a similar washing procedure was repeated about five times using the 0.1 M NaCl solution until the pH of the supernatant solution equaled that of the 0.1 M NaCl solution (pH 5.5). After the last wash, the clay was redispersed in 0.1 M NaCl and stored in suspension prior to the spectroscopic analysis.

The clay preparation procedure used for the KGa-1 kaolinite sample was similar to the procedure described by Johnston et al. (Reference 10). Two hundred grams of the untreated KGa-1 clay were placed in 1 liter of distilled, deionized water and dispersed for size fractionation by adjusting the pH to 9.5 by the addition of few drops of 0.01 M NaOH. The kaolinite-suspension was size fractionated immediately by centrifugation and the fraction have an e.s.d. of <2.0 μm was collected. The suspension then was flocculated by the addition of 1 liter of 0.0001 M HCl in 1.0M NaCl. To separate the supernatant solution from the flocculated clay, the suspension was centrifuged at a relative centifugal force of 700. The supernatant solution was then decanted and its pH value measured. The kaolinite samples were redispersed manually into 1 liter of the 0.0001 M HCl/1.0 M NaCl solution, and the washing procedure was repeated until the pH value of the supernatant solution equaled that of the washing solution (pH 3) this objective typically required five washed. At this point in the procedure, the clay was redispersed into 1 liter of 0.01 M NaCl, and the above procedure was repeated five more times. The treatment was adequate to raise the pH of the supernatant solution to 5.5. The final step of the procedure consisted of redispersing the clay into 0.01 M NaCl and adjusting the volume of the flocculated suspension such that a clay concentration of 20 percent (w/w) was obtained.

C. DESCRIPTION OF THE BOMEN DA3.10 FOURIER TRANSFORM SPECTROMETER

FT-IR spectra were obtained on a Bomem DA3.10 Fourier transform spectrometer. The DA3.10 spectrometer utilizes a Michelson interferometer with the beamsplitter positioned at a 30-degree angle to the optical axis. A sixty degree field-of-view Infrared Associates broad-band, liquid nitrogen cooled, mercury-cadmium-telluride (MCT) detector fitted with a KRS-5 infrared window was used for these FT-IR studies. The active area of the detector element was 1.032 mm_2. The measured D* value of the detector was 3.13×10^9 cm $Hz^{0.5}$ and the cutoff wavenumber was 400 cm^{1-} (25 microns). A midinfrared, watercooled, ceramic silicon carbide source was used for the mid-IR region and a visible quartz tungsten halogen source, mounted inside the spectrometer, was used for sample alignment.

The optical resolution used in these studies ranged between 2.0 and 0.5 wavenumbers. A preliminary study showed that the spectra of kaolinite were not instrument-limited for nominal resolution values of 0.5, 1.0, 2.0 cm-1. At 4.0 cm^{-1} resolution, however, the spectrum of kaolinite was instrument-limited. The FT-IR spectra of the SAz-1 Cheto montmorillonite sample were not instrument limited at a resolution of 2.0 cm^{-1}. A Hamming apodization function was used to weight the cosine wave interferograms. Initially, a low-resolution double-sided interferogram was collected and the phase angle deviation from zero of the interferograms was determined using the Forman method (References 13-15). The phase correction determined from the double sided interferogram was stored in the HSVP. Subsequently, all single sided, high resolution interferograms were corrected using these stored values. Typically, 16000 data points collected per single sided interferogram with approximately 900 data points collected before the centerbrust. Programmable low-pass and high-pass analog filters were used to optimize the signal-to-noise ratio. A low pass cut-off filter of 20 KHz, and a high-pass cut-off filter of 2 Nz (3 db cut-off frequencies). Interferograms were collected with a moving mirror velocity of 0.5 cm/sec which corresponded to a sampling frequency of 15.8 KHz. The dynamic range of the analog-to-digital converter was 16 bits and the word length of the Vaxstation-II computer was 32 bits. One sample point was

collected per laser fringe with a resolution of 16 bits per sample. Typically, 256 scans were coadded for the sample and reference files. The total measurement time for coadding 256 scans was 200 seconds for 1 cm^{-1} resolution and 600 seconds for 0.5 cm^{-1} resolution. No smoothing or interpolation algorithms were used.

The sample compartment of the Bomem DA3.10 was operated under a reduced pressure of 0.05 torr to remove interferences from H_2O and CO_2 and other vapor phase constituents. The sample cover access plates were modified to accommodate two 3/8" MDC quick-disconnect vacuum tube feed-throughs which provided a connection through the vacuum wall of the spectrometer from the vacuum manifold to the CE-TR cell in the sample compartment.

D. DESCRIPTION OF THE VAXSTATION-II DATA ACQUISTION SYSTEM

The data acquisition system for the Bomem DA3.10 spectrometer was a Digital Equipment Corporation (DEC) Vaxstation-II computer. The Vaxstation-II computer consists of a μVAX-ii cpu with a dedicated high-resolution, bit-mapped graphic display terminal. The Bomem DA3.10 spectrometer is connected directly to a high-speed vector processor (HSVP) through a dedicated 50-line parallel interface. The Vaxstation-II communicated with the HSVP through a National Instruments General Purpose Interface Bux (GPIB) Card (Model No. GPIB11V-2) which was resident on the Q-bus of the Vaxstation-II. The initial data acquisition system used to collect data from the Bomem was a DEC PDP 11/23 computer (which also supports the Q-bus). The use of the PDP 11/23 was limited for this application because of the 16-bit word-length and address restriction, limited storage capability, and the extremely slow display output which resulted from the 9600 baud serial-throttle of the VT-240 terminal. The slow display and low resolution of the VT-240 terminal, in particular, were serious limitations of the PDP 11/23. In terms of overall computational performance, the Vaxstation-II (μVAX-II) system is ~ 20 times faster than the PDP 11/23. Bomem does not support their software on the Vaxstation-II, thus, the Bomem Fortran-77 and assembly-level (Macro-11) codes were modified and transferred to the Vaxstation-II. Most of the routines could be transferred directly to the Vaxstation-II

with little, or no, modification; however, the HSVP routines had to be re-written altogether because of the significant differences between the VAX/VMS and RT-11 operating systems.

In addition to the HSVP routines, new codes were developed to support the VR260 graphic display terminal. The Vaxstation-II software provided two environments for the development of the graphic display codes: the VAX GKS run-time library of graphical functions that are defined by the ANSI X3.124-1985 and ISO 7942-1985 Graphic Kernel System (GKS) standards, or the device dependent MicroVMS Workstation Graphics. The VAX GKS environment was chosen over the latter because GKS is supported on a number of different machines (e.g., DEC, IBM, HP etc.) and support a number of different graphic output formats including the Tektronix 4010/4014, DEC VT-125, DEC Vaxstation-II VR260, LVP16, HP7470, HP7475A, LN03, and GKS Metafile formats. The graphic display codes were developed on the Vaxstation-II operating under the VAX/VMS Version 4.5 operating system using V3.1 of the MicroVMS workstation software. The VAX GKS V2.0 run-time library of graphical functions were called form VAX Fortran-77 V4.

E. DESCRIPTION OF THE CONTROLLED-ENVIRONMENT-TRANSMISSION CELL AND MANIFOLD

The controlled-environment-transmission (CE-TR) cell was a modified 10 cm pathlength cell fitted with two Kontes teflon stockcocks, and two 49 mm x 3 mm ZnSe windows using Viton o-rings. The kontes teflon stockcocks were modified at the glass shop to accommodate two Lab-Crest Model 571-190 9 mm Solv-Seal® joints. The Solv-Seal joint system incorporates a TFE seal with two Viton o-rings and can be pumped down to 10^{-8} torr. Clay films were held in place in the CE-TR cell using a TFE holder which allowed the films to be mounted at 90 or 60 degree angles of incidence to the modulated IR beam. The CE-TR cell was mounted on a Newport Research Corporation (NRC) Model 460-XYZ-DM translation stage which was used to position the cell in the spectrometer.

The vacuum system consisted of a 195 liter-minute^{-1} mechanical pump, 3-angstrom molecular sieve trap, three way isolation valve, water-cooled Edwards Eiffstak 63 oil-diffusion pump (135 liter-sec^{-1}), Edwards Penning

& Pirani gauge-head assembly, ISO-63 flange to KF-40 flange adapter, KF-40 isolation valve, KF-16 relief valve, 1 meter KF-40 stainless steel bellows, glass liquid nitrogen trap fitted with Lab-Crest Model 571-190 15 mm Solv-seal joints, and a five-place Teflon®/glass Airless-ware manifold fitted with Lab-Crest Model 571-190 9 mm Solv-Seal® joints. Because all of the glass joints were terminated with the Lab-Crest Solv-seal joints, the vacuum manifold was highly modular. An Edwards Model 1005 controller fitted with two Pirani gauge heads and one Penning gauge were used to monitor the pressure in the CE-TR cell and vacuum manifold in the 760 torr to 10^{-7} torr range.

F. LOW TEMPERATURE FT-IR STUDIES

Low temperature FT-IR spectra were obtained on the Bomem DA3.10 spectrometer using an air cooled Air Products CS-202 cryogenic refrigeration system. The CS-202 expander module was mounted in a non-standard vacuum flange in the sample compartment of the DA3.10 spectrometer. The expander module was connected to an external medium/high vacuum pumping system (10^{-8}torr). A 25mm x 2mm ZnSe window was mounted in the Air Products DMX-1 sample holder using indium gaskets. The vacuum shroud was fitted with 2 49mm x 4mm ZnSe windows using Viton o-rings. During the operation of the cryogenic refrigeration system, the pressure in the expander module was maintained below 5 x 10^{-6} torr. The temperature of the sample holder was measured using a Chromel/Gold 0.07 Atomic % Iron thermocouple connected to an Air Products Model APD-E digital temperature controller. Temperature of the sample holder was regulated, using the APD-E controller and a 20-watt proportional-plus-reset-plus-rate controller.

G. RESULTS

Figure 26 shows the controlled-environment transmission (CE-TR) FT-IR spectrum of a thin deposit of KGa-1 kaolinite on a ZnSe window in the 400 cm^{-1} to 4000 cm^{-1} region. Expanded plots of this spectrum are shown in Figures 27 and 28 and the observed band positions are tabulated in Table 13. Preliminary FT-IR studies showed that the KGa-1 kaolinite

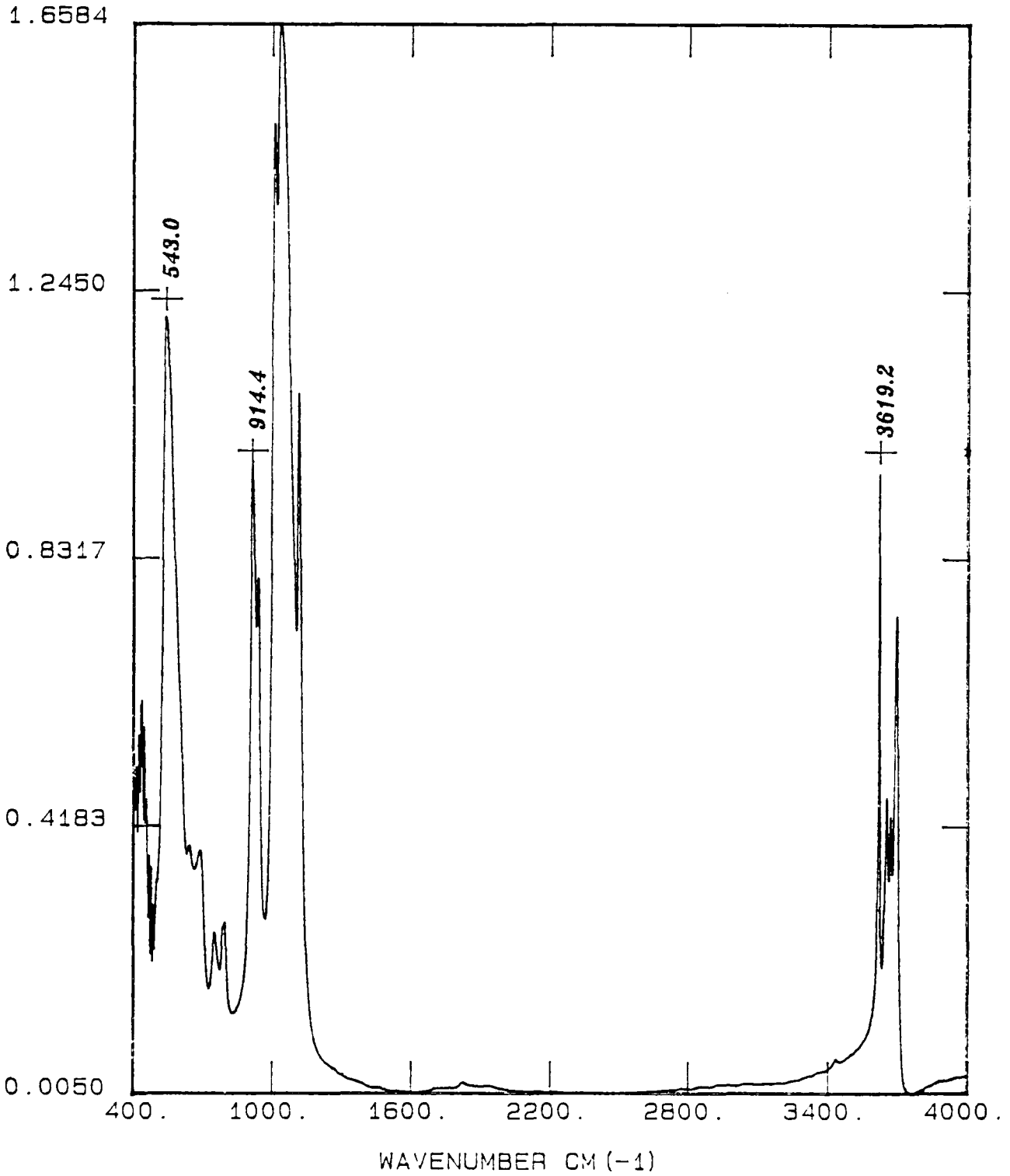


Figure 26. Controlled Environment Transmission (CE-Tr) FT-IR Absorbance Spectra of KAa-1 Kaolinite in the 500 cm^{-1} to 4000 cm^{-1} Region at 300 K.

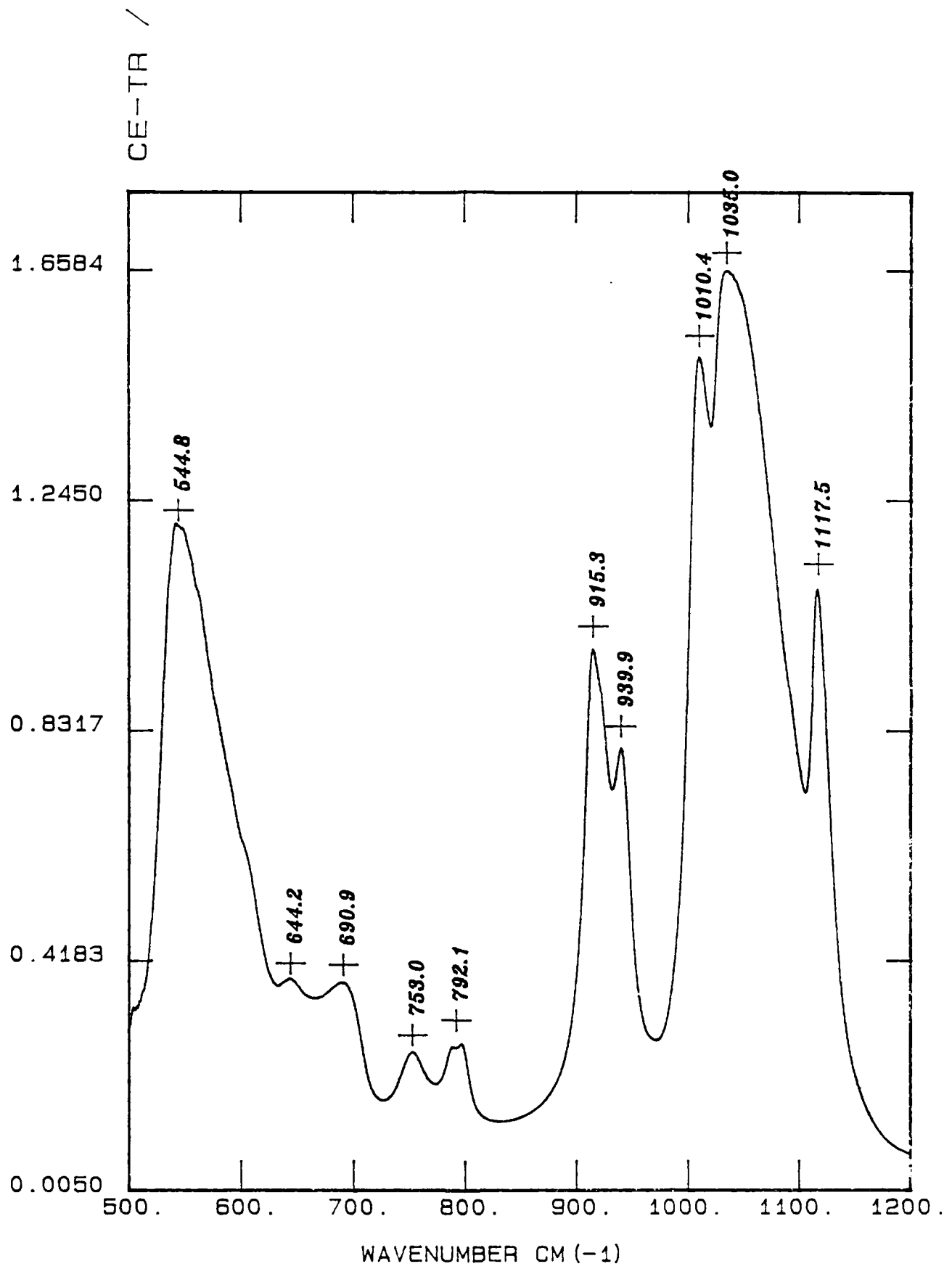


Figure 27. Ce-Tr-Ft-IR Absorbance Spectra of KGa-1 Kaolinite in the 500 cm^{-1} to 1200 cm^{-1} Region at 300 K.

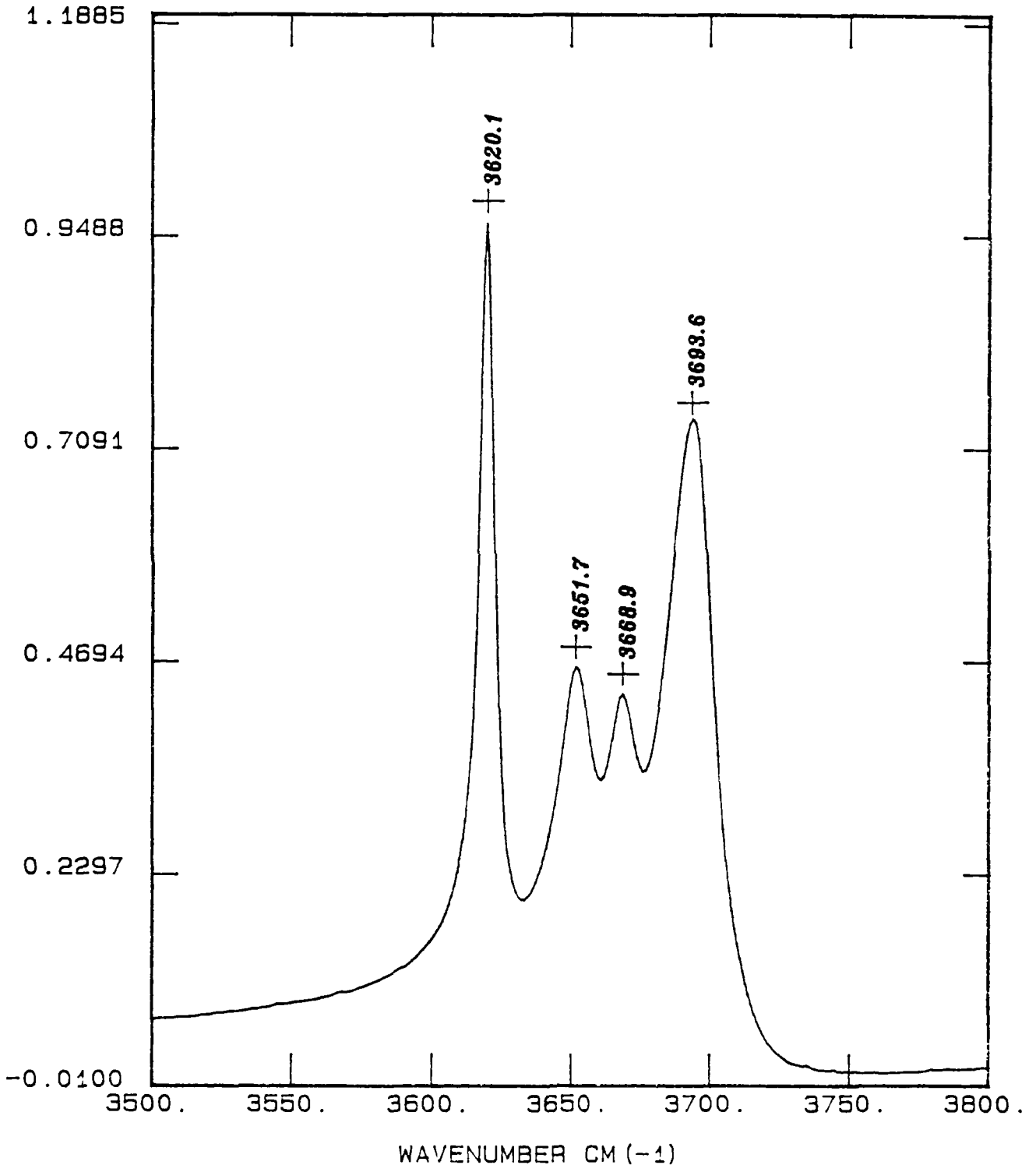


Figure 28. CE-TR-FT-IR Absorbance Spectra of KGa-1 Kaolinite in the 3550 cm^{-1} to 3750 cm^{-1} Region at 300 K.

TABLE 13. BAND ASSIGNMENTS OF KGa-1 KAOLINITE.

544	Si-O-Al skeletal vibrations
606.0	Si-O-Al skeletal vibrations
643.4	
690.1	-OH deformation - gibbsite layer
787.8	-OH deformation - gibbsite layer
799.7	
915.3	Al-O-H deformation, Inner-Surface hydroxyl
939.9	Al-O-H deformation, Inner Hydroxyl
1009.6	Si-O stretch
1034.2	Si-O stretch
1116.6	Si-O stretch
3620.1	Al-O-H stretch, Inner hydroxyl group
3652.1	Al-O-H stretch, Inner-Surface hydroxyl group
3668.9	Al-O-H stretch, Inner-Surface hydroxyl group
3693.9	Al-O-H stretch, Inner-Surface hydroxyl group

spectra were not instrument limited at 0.5 and 1.0 cm^{-1} nominal resolution values. At 2.0 cm^{-1} a nominal resolution value of 1.0 cm^{-1} was chosen for this study. The positive-slope of the baseline in the spectrum of the KGa-1 kaolinite deposit shown in Figure 1 is a reproducible feature of FT-IR spectra of kaolinite deposits on IR window materials, ZnSe was used in this study.

The derivative shaped baseline in the v(O-H) region is an artifact resulting from the Reststrahlen effect (Reference 16). This behavior results from the variation of the refractive index of the sample near the absorption bands of the clay. No baseline correction or smoothing algorithms were used to manipulate the spectral data presented in Figures 26 to 28. One advantage of diffuse reflectance sample presentation over the transmission sample presentation method employed in this study is that the Reststrahlen effect is absent using the diffuse reflectance method (Reference 17).

An expanded plot of the KGa-1 kaolinite absorbance FT-IR spectrum in the 500 to 1200 cm^{-1} region is shown in Figure 27. There is general agreement in the literature that the bands at 1010, 1034, and 1117 cm^{-1} are Si-O stretching bands. The frequencies of these bands, the 1117 cm^{-1} band in particular, are sensitive to the size and shape of the particles, and to the orientation of the particles with respect to the IR beam. Selective deuteration (Reference 18-19) studies have shown that the 915 and 940 cm^{-1} bands are assigned to the Al-O-H deformation bands of the inner-surface, and inner hydroxyl groups of kaolinite, respectively. Farmer et al. (References 20-21) have assigned the 690, 753 and 799 cm^{-1} bands to hydroxy groups of the gibbsite layer of kaolinite. The band assignments for the lower frequency modes of kaolinite have not been assigned; however, Stubican and Roy (Reference 22) have suggested that the 544 band is a Si-O-Al skeletal vibration.

Four well-resolved bands are observed in the FT-IR spectrum of kaolinite in the hydroxyl stretching region at 3620, 3652, 3669, and 3694 cm^{-1} (Figure 28). For comparison, the Raman spectra of KGa-1 kaolinite in the 3600 to 3725 cm-1 regions are presented in Figure 29-31. The FT-IR spectrum (Figure 28) and the Raman spectra (Figure 31) of the KGa-1 kaolinite sample are similar. The Raman spectra of the Mesa Alta

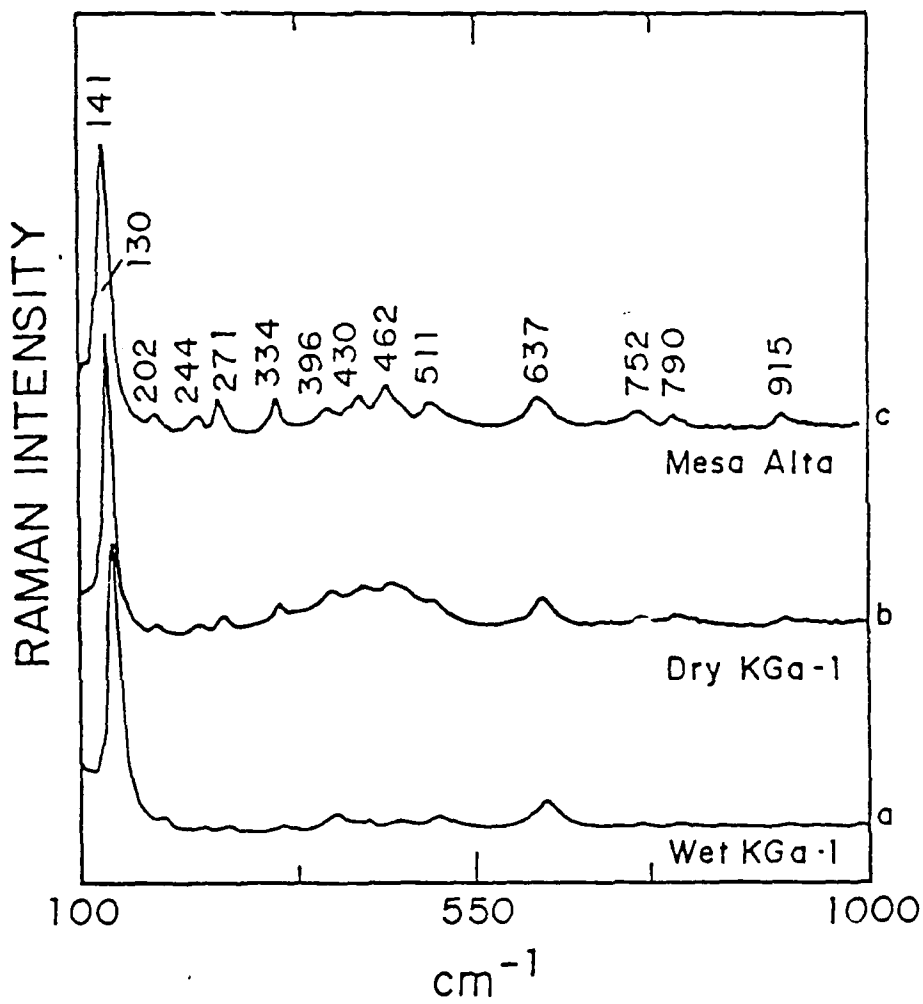


Figure 29. Raman Spectra of KGa-1 Kaolinite in Aqueous Suspension (A), Dry KGa-1 Kaolinite (B), and Dry Mesa Alta Kaolinte (C) in the 100 to 1000 cm^{-1} Region.

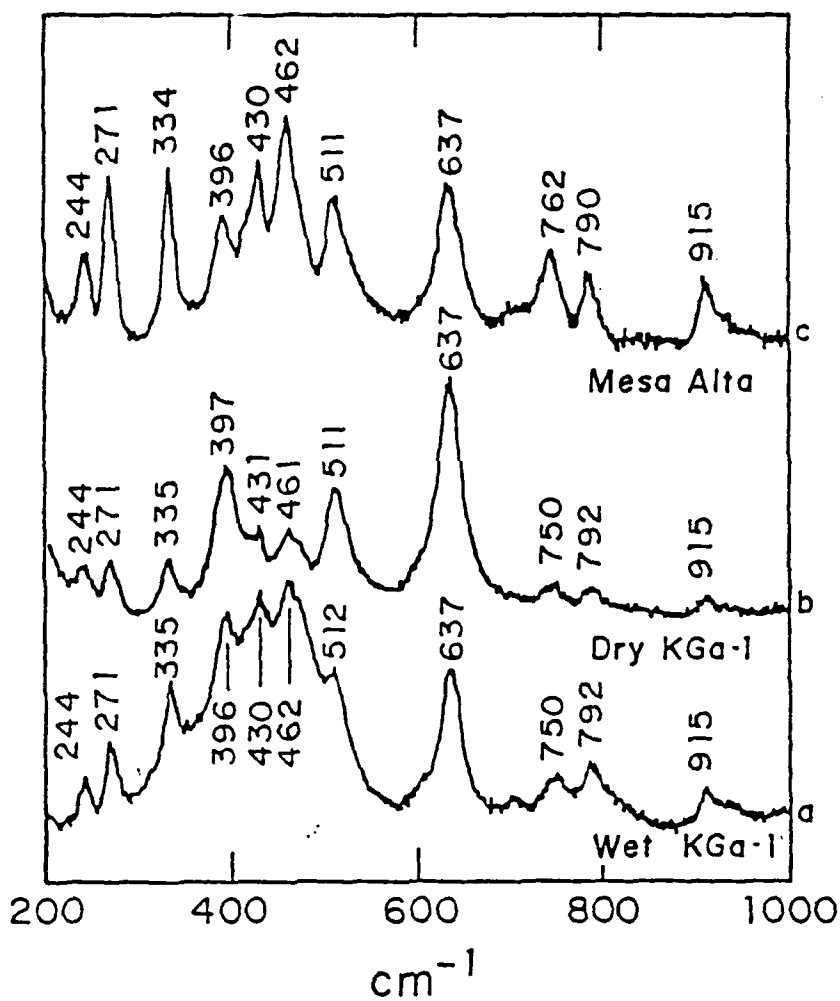


Figure 30. Raman Spectra of KGa-1 Kaolinite in Aqueous Suspension (A), Dry KGa-1 Kaolinite (B), Dry Mesa Alta Kaolinite (C) in the 100 to 2000 cm^{-1} Region.

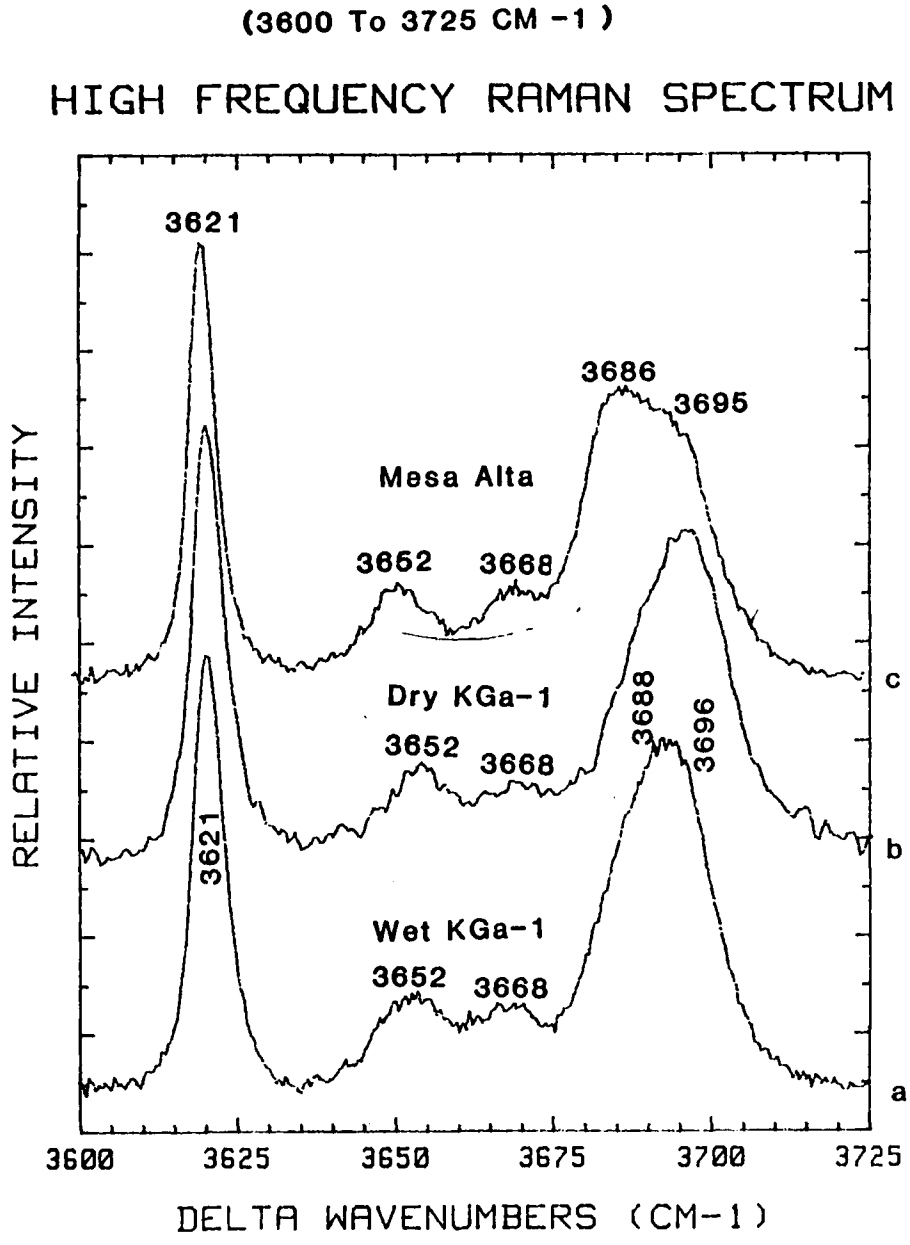


Figure 31. Raman Spectra of KGa-1 Kaolinite in Aqueous Suspension (A), Dry KGa-1 Kaolinite (B), and Dry Mesa Alta Kaolinite (C) in the 3500 to 3800 cm^{-1} Region.

kaolinite sample are similar. The Raman spectrum of the Mesa Alta kaolinite sample, however, shown at the top of Figure 31, has a fifth hydroxyl stretching band which is Raman-active spectrum but not in the IR-active (Reference 10). The observation of five Raman O-H stretching bands was first reported by Wiewora et al (Reference 23) for a Keokuk kaolinite sample.

Band positions obtained from the CE-TR spectrum (Figures 26-28) are shown in Table 13 and are in good agreement with the dispersive-IR and Raman literature values for kaolinite (Reference 9). For comparison, the Raman spectra of KGa-1 kaolinite in the 3600 to 3725 cm^{-1}, 100 to 1000 cm^{-1}, and 200 to 1000 cm-1 regions are shown in Figures 29, 30. There is a strong correspondence between the Raman and IR spectra of kaolinite in the hydroxyl stretching region (Figure 31). What is not apparent from this comparison, however, were the large differences in scan collections times; for a 110 cm^{-1} resolution Raman scan of kaolinite using a typical dwell time of 5 sec $step^{-1}$ in the 100 to 4000 cm^{-1} region would require 325 minutes (5.5 hours) of sample collection time. In contrast, a comparable SNR could be obtained in approximately 1 minute using CE-TR FT-IR. Thus, CE-TR FT-IR methods are characterized by a much higher sensitivity than Raman techniques for studies of clay minerals. In the hydroxyl stretching region of kaolinite, CE-TR provides a considerable advantage over Raman methods considering the similarity of the spectra in this region and the much longer scan times required for Raman spectra. In contrast to the good overall agreement between the Raman and CE-TR spectra in the hydroxyl stretching region, there is poor correspondence between the spectral band positions and relative intensities of the Raman (Figures 29, 30) and CE-TR (Figure 27) spectra in the low frequency region. Thus, a more complete characterization of the lower frequency vibrational modes of kaolinite is obtained using a combined application Raman and FT-IR methods (Reference 22).

The crystal structure of kaolinite projected onto the (100) plane is shown in Figure 32. This drawing illustrates the two distinct types of hydroxyl groups that reside within the crystal structure of kaolinite; the inner hydroxyl "sandwiched" between the octahedral and tetrahedral layers of the clay lattice, and the inner-surface hydroxyl groups located on internal surface of the "gibbsite-like" Aluminum octahedral layer. The 3620 cm^{-1} band has been conclusively assigned to the inner hydroxyl group of kaolinite. This hydroxyl group is highly resistant to isotopic exchange with deuterium and to dehydroxylation at elevated temperatures relative to the other hydroxyl stretching bands because of its recessed location with the kaolinite structure.

H. LOW TEMPERATURE FT-IR STUDIES OF KGa-1 KAOLINITE[2]

One of the objectives of the research effort was to obtain low temperature FT-IR spectra of the kaolinite-hydrazine complex. Low temperature FT-IR spectra of kaolinite, or any other clay mineral, have not been reported in the literature. Consequently, FT-IR spectra of kaolinite were obtained as a function of temperature to assist in assigning low temperature features of the kaolinite-hydrazine FT-IR spectrum. The lower frequency bands of kaolinite in the 600 to 1200 cm^{-1} region are presented in Figure 33 as a function of temperature. A scatter plot of the relative shift of the low frequency bands relative to the root temperature frequency positions is presented in Figure 34. In general, only a minor increase in frequency is observed for the lower frequency bands. The observed increase in frequency upon cooling is related to the thermal contraction of the unit cell of kaolinite. Figure 35 shows the relative frequency response of the Al-O-H deformation models as a function of temperature. The shift in frequency of the 915 cm^{-1} band as a function of temperature is generally the same as that observed for the other low frequency modes; the frequency of the band increases upon cooling resulting from the temperature induced contraction of the clay lattice. The behavior of the 940 cm^{-1} band is unique, however, in that the observed frequency does not increase at lower temperatures.

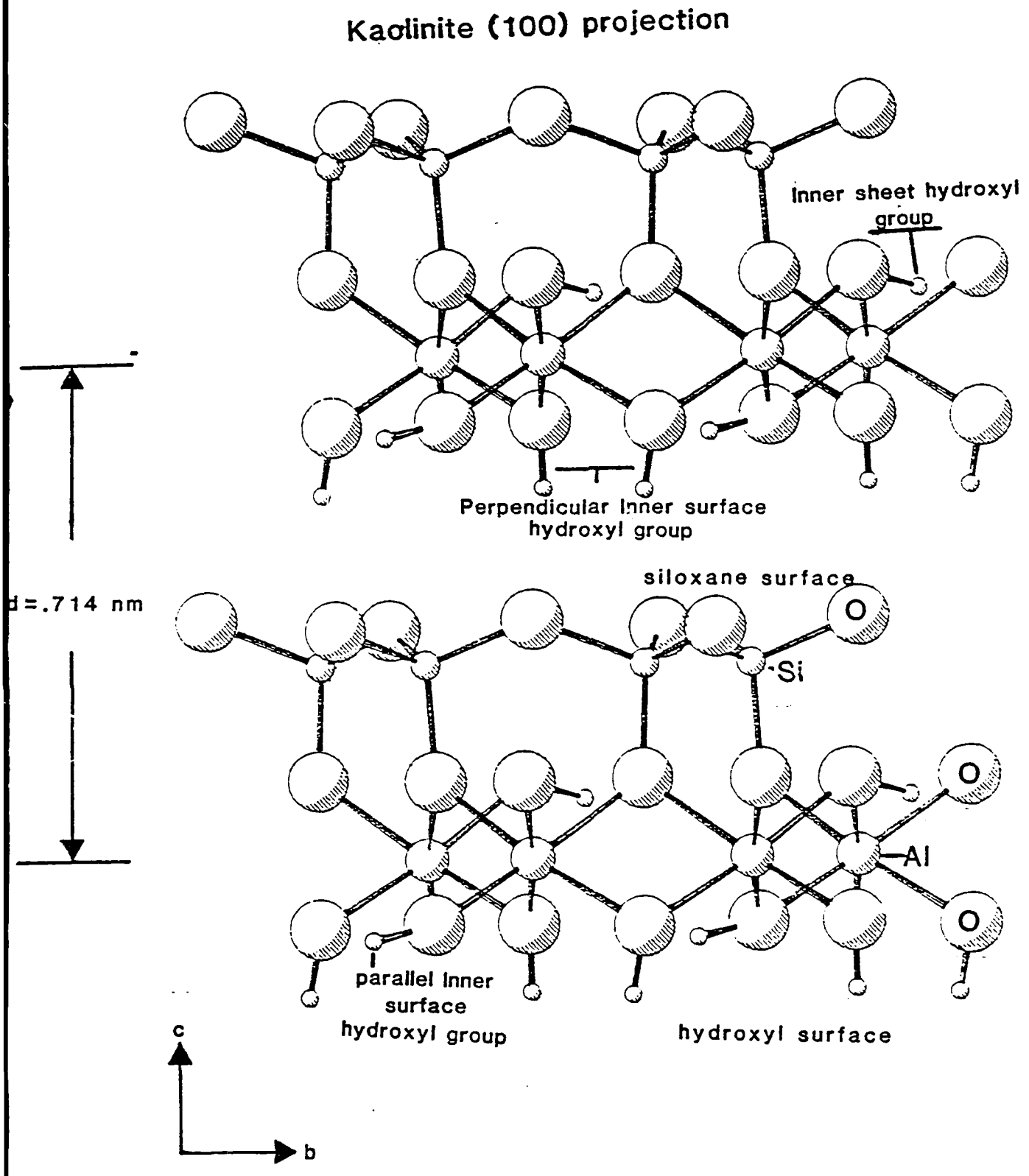


Figure 32. Crystal Structure of Kaolinite Projected onto the [100] Plant Showing the Location of the Inner and Inner-surface Hydroxyl Groups.

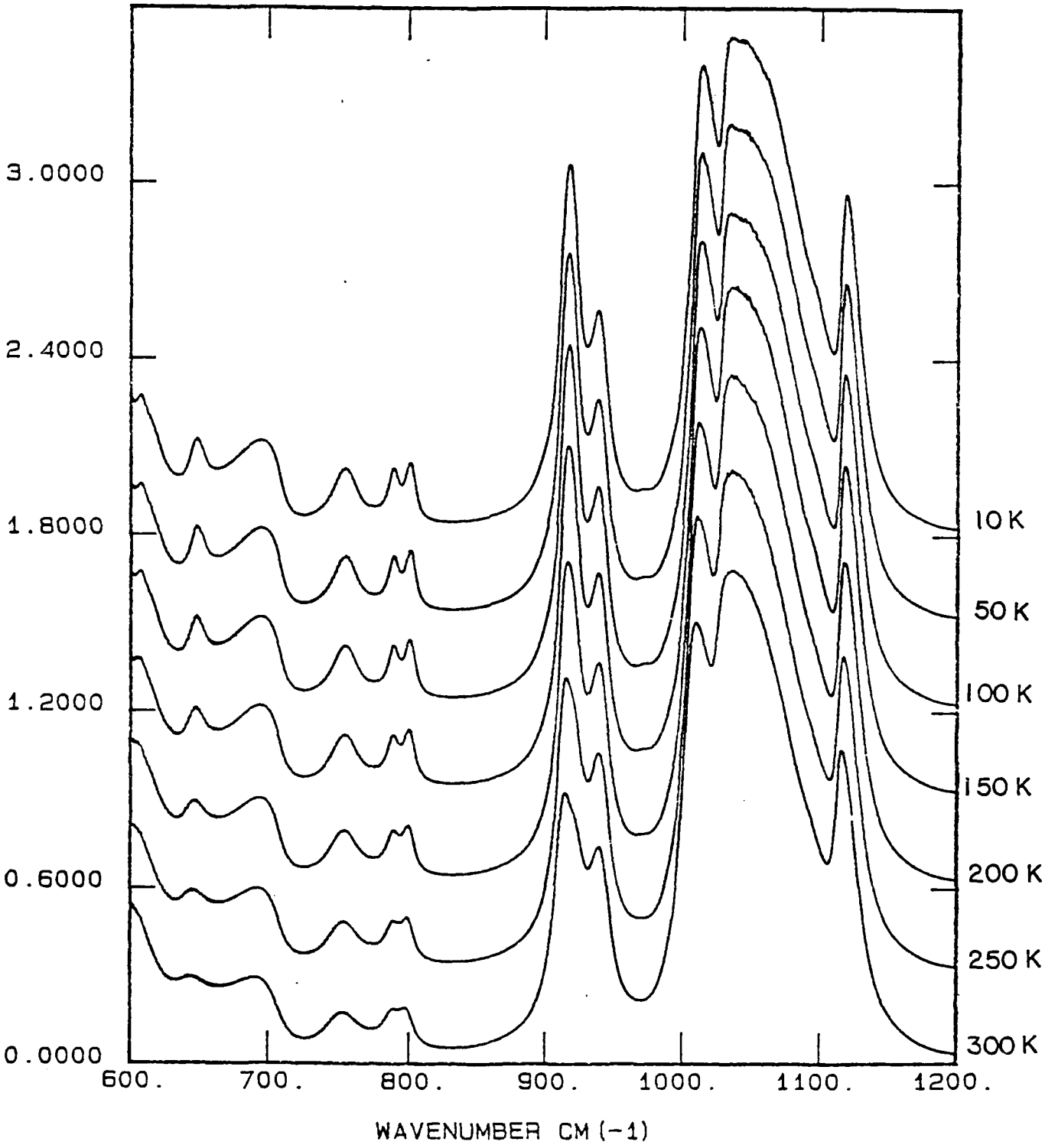


Figure 33. Temperature Dependence of the FT-IR Absorbance Spectra of KGa-1 Kaolinite in the 600 to 1200 cm^{-1} Region.

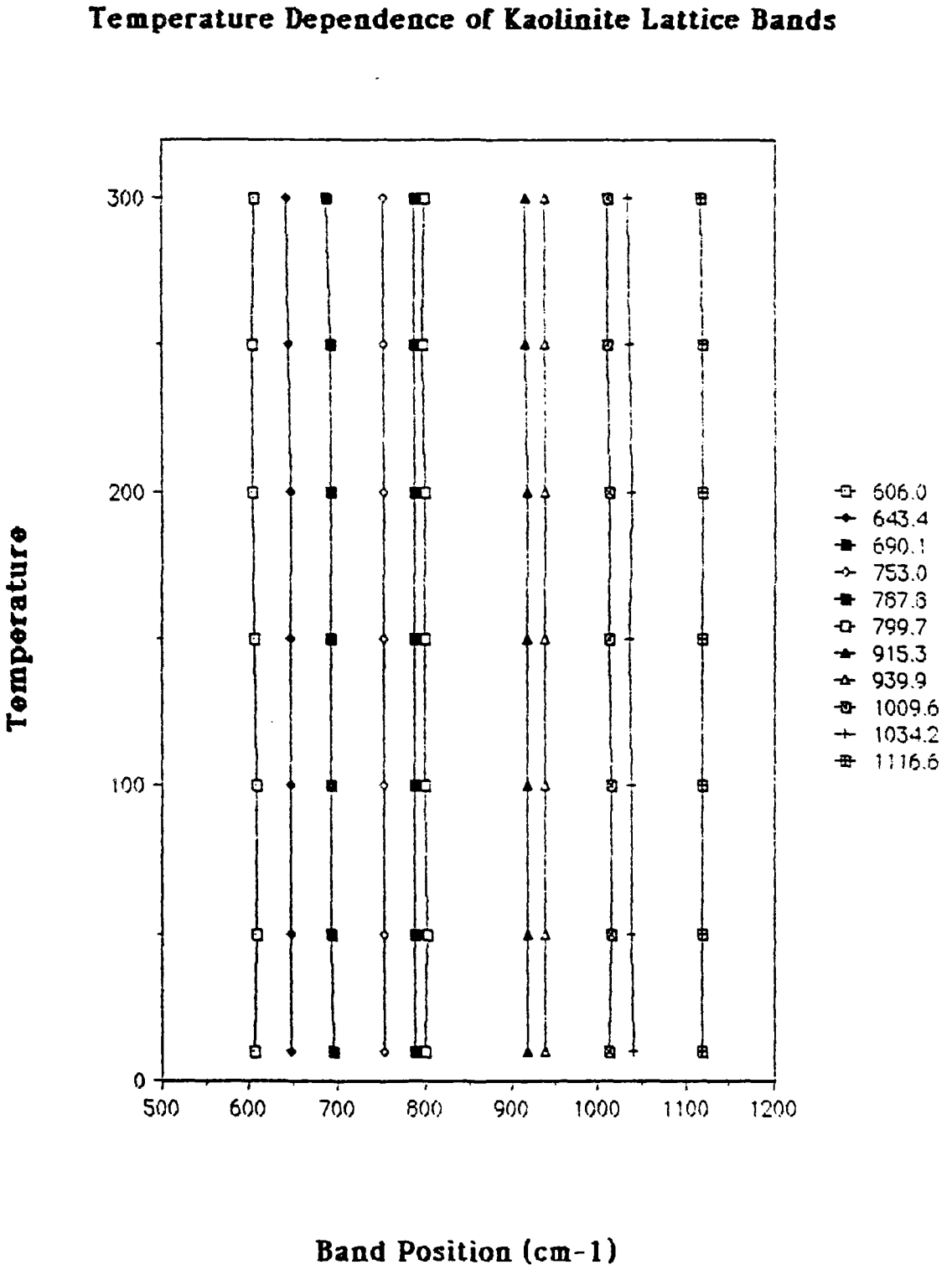


Figure 34. Plot of the Lower Frequency IR-active Modes of Kaolinite as a Function of Temperature.

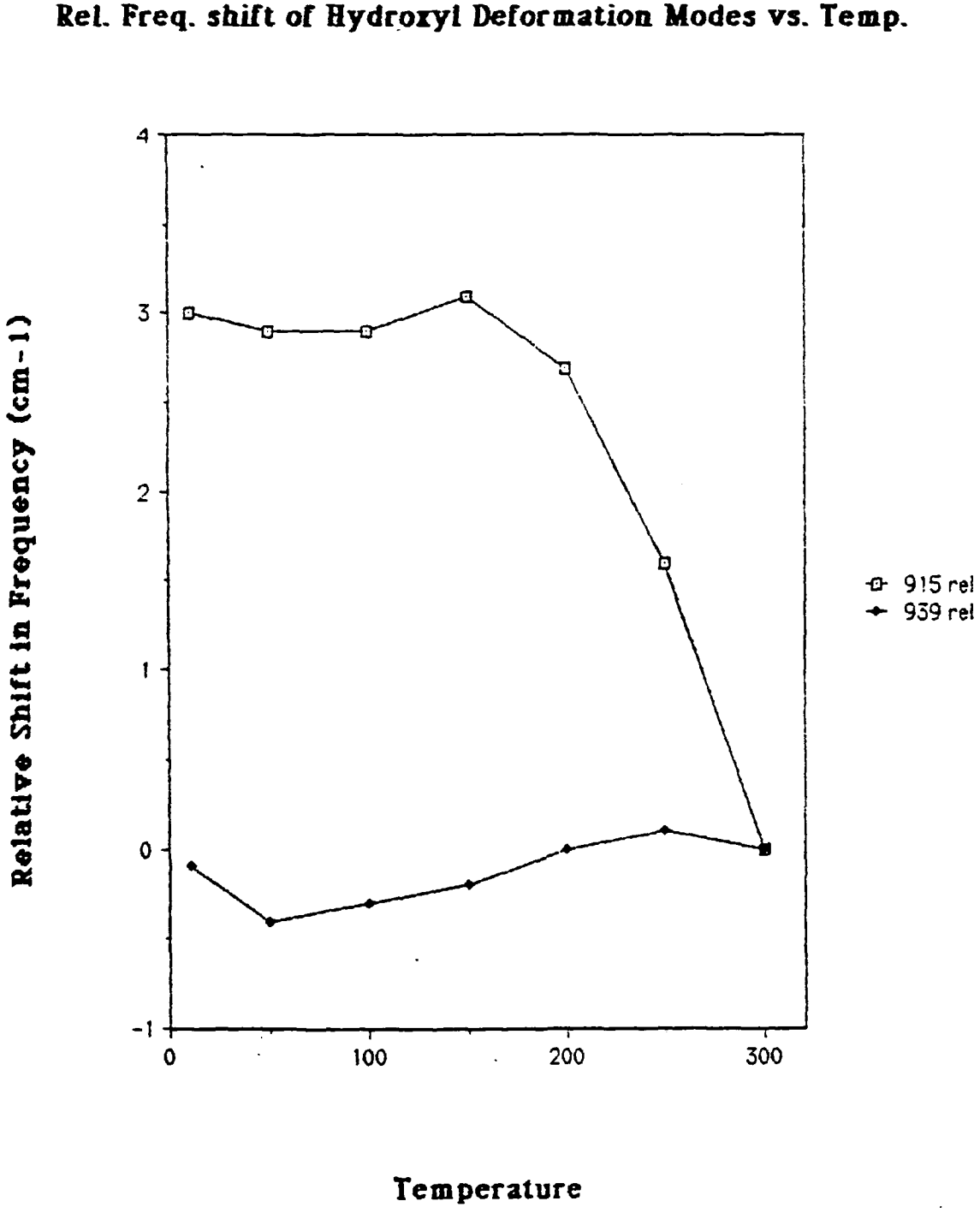


Figure 35. Shift in Frequency of the Hydroxyl Deformation Modes of Kaolinite Relative to Their Band Positions at 300 K.

The temperature dependence of the IR-active hydroxyl stretching bands is shown in Figure 36. The bank positions of the observed bands are presented in Table 14 as a function of temperature. Prost (Reference 24) reported the IR spectrum of kaolinite cooled to 150 K recently. The spectra reported in Figure 36 represent the first reported FT-IR spectra of kaolinite cooled to near liquid helium temperatures. One of the main advantages in vibrational spectroscopy of cooling samples is that the natural linewidths associated with vibrational transitions are temperature-dependent; i.e., thus, greater resolution of vibrational structure is often realized at lower temperatures. The low temperature spectrum of kaolinite obtained at 10 K shows clearly the presence of several "new" vibrational features which are not present or resolved at room temperature. These "new" low-temperature features of the kaolinite spectrum have not been assigned as yet; however, the data presented in Figure 36 do show that increased resolution of the hydroxyl stretching bands of kaolinite is realized at near liquid helium temperatures.

Figure 37 presents the temperature dependence of the hydroxyl stretch bands and Figure 38 shows the relative shift in frequency of these bands compared to the room temperature band position. The observed shifts are in agreement with the direction and magnitude of the shifts reported in the literature (References 24-25). No explanation has been provided, as yet, to account for the blue shift of the 3620 cm^{-1} band and the observed red shift of the inner-surface hydroxyl bands. The average net <u>increase</u> in frequency of the inner-surface hydroxyl groups observed upon cooling from 300 K to 10 K is 13 cm^{-1}; in contrast, the frequency of the 3620 cm^{-1} band was observed to <u>decrease</u> by -4 cm^{-1}. These observed opposite shifts of the hydroxyl stretching bands may provide a useful spectroscopic method for identifying OH stretching bands in the future.

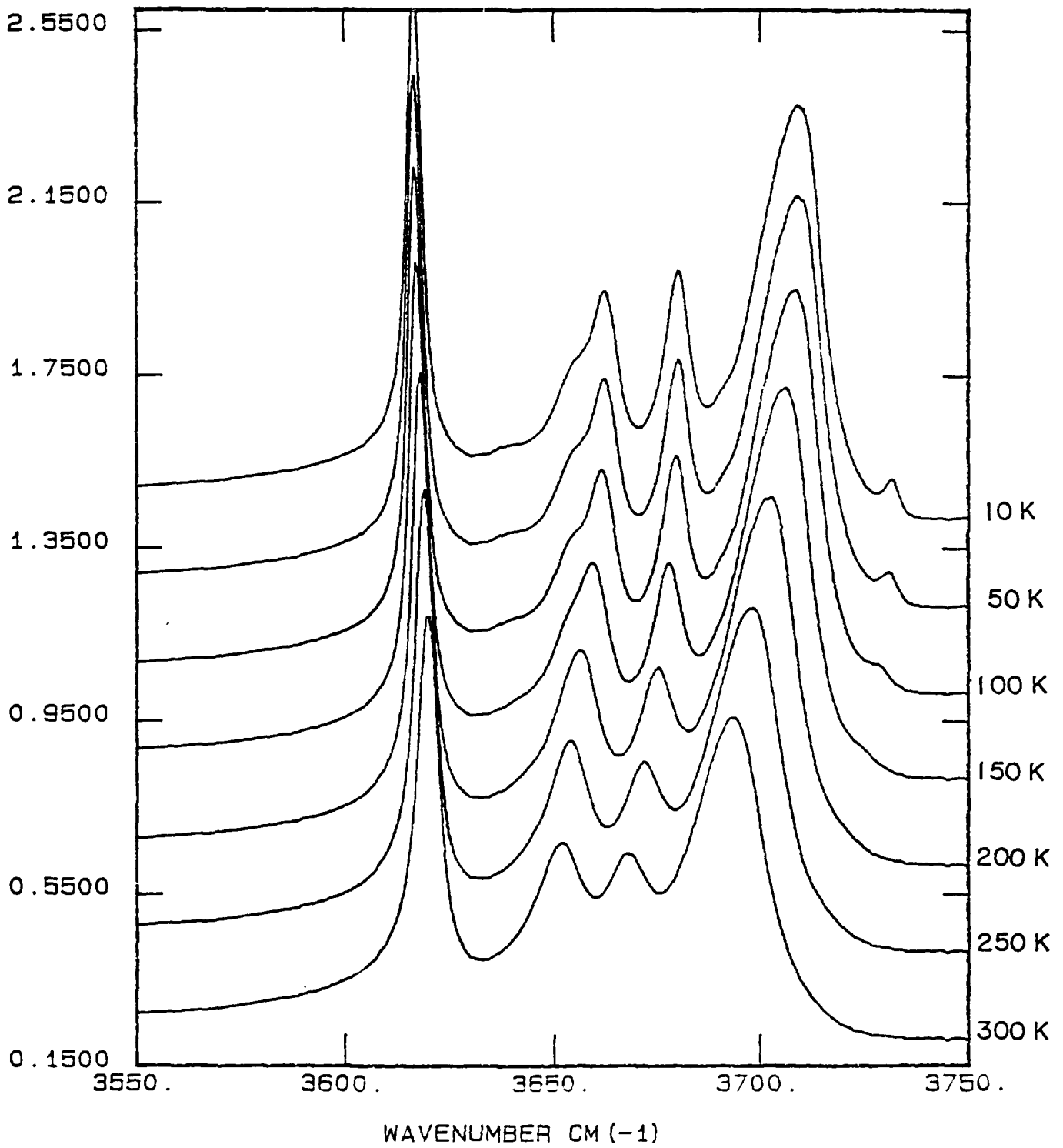


Figure 36. Temperature Dependence of the FT-IR Absorbance Spectra of KGa-1 Kaolinite in the 3550 to 3750 cm^{-1} Region.

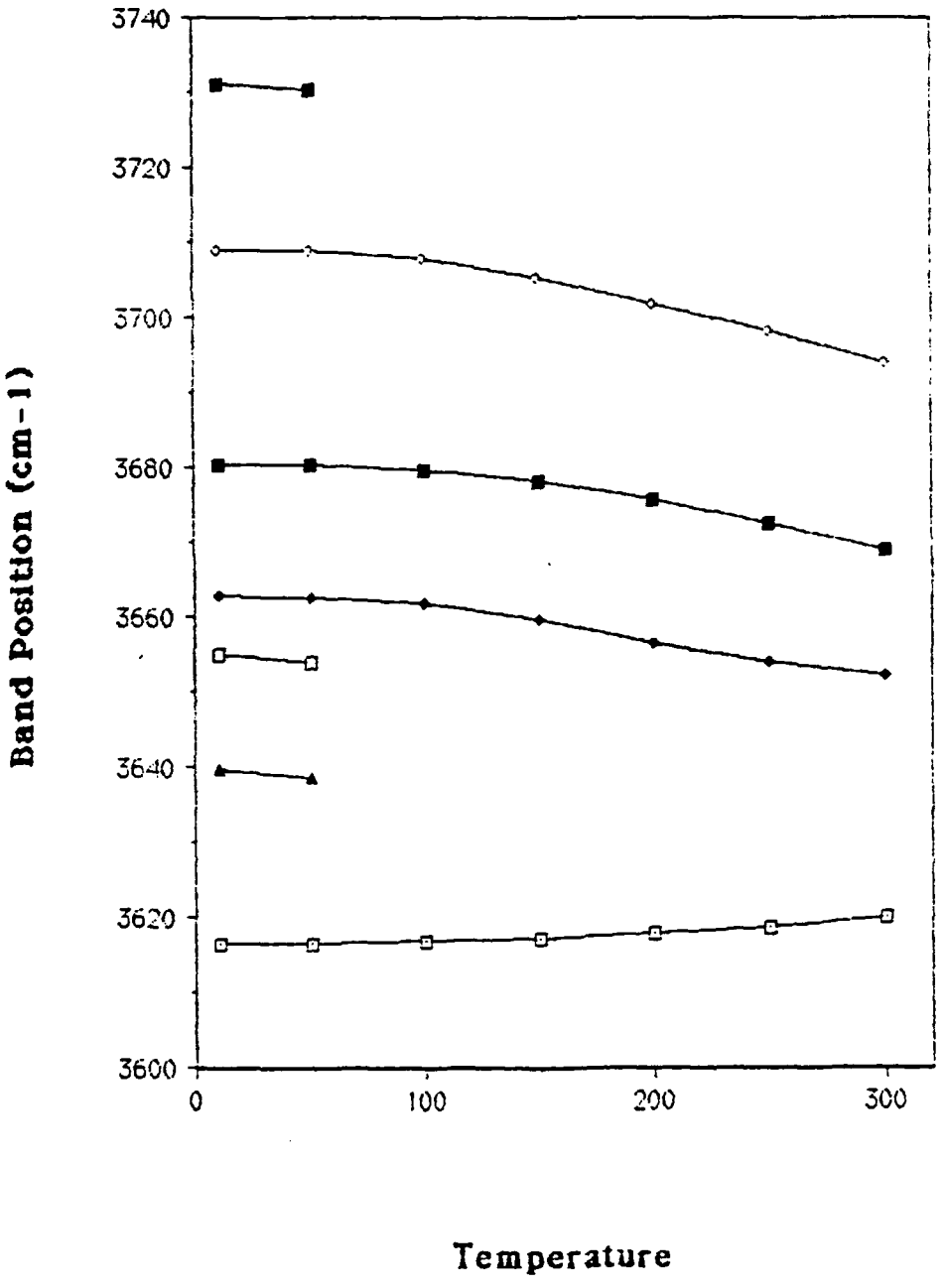


Figure 37. Plot of the Band Positions of the IR-active Hydroxyl Stretching Bands as a Function of Temperature.

:mperature induced shift in band position of hydroxyl stretching bands

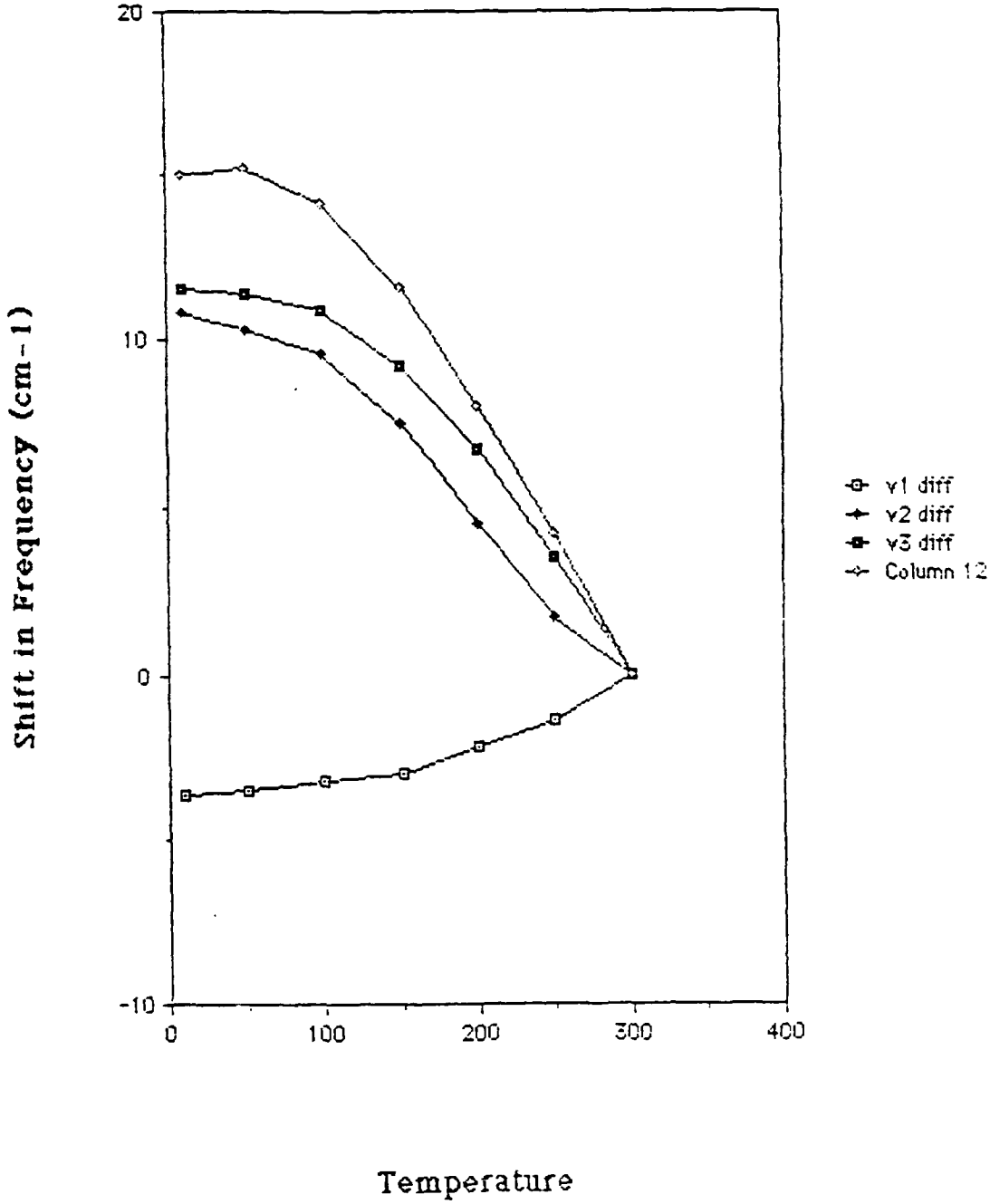


Figure 38. Shift in Frequency of the Hydroxyl Stretching Modes of Kaolinite Relative to Their Band Positions at 300 K.

TABLE 14. BAND POSITIONS OF KGa-1 KAOLINTE AT 300 K, 250 K, 200 K, 150 K, 50 K AND 10 K.

606.0	604.3	603.9	605.9	607.5	608.1	607.4
643.4	645.8	646.8	647.7	647.8	649.0	648.4
690.1	693.5	693.1	694.8	694.4	649.7	695.8
753.0	753.9	754.6	754.4	754.4	754.7	754.4
787.8	789.1	789.3	788.8	789.6	789.6	788.6
799.7	798.1	799.8	800.4	800.0	801.6	800.8
915.3	916.9	918.0	918.4	918.2	918.2	918.3
939.9	940.	939.9	939.7	939.6	939.5	939.8
1009.6	1011.6	1011.8	1013.8	1014.3	1014.2	1014
1034.2	1036.9	1037.3	1036.8	1039.1	1038.3	1039.5
1117.1	1118.4	1118.4	1118.4	1118.8	1118.8	1119
3620.1	3618.7	3617.9	3617.1	3616.9	3616.6	3616.5
					3638.5	3639.5
					3654.0	3655.01
3652.1	3653.8	3656.6	3659.6	3661.7	3662.4	3662.9
3668.9	3672.4	3675.6	3678.1	3679.8	3680.3	3680.4
3693.9	3698.1	3701.9	3705.5	3708.0	3709.1	3708.9
					3730.3	3731.2

I. FT-IR STUDIES OF THE KAOLINITE-HYRDRAZINE INTERCALATION COMPLEX

Expansion of the kaolinite lattice upon intercalation by hydrazine is by a plot of the X-ray diffraction d_{001} reflections of kaolinite and of the KH complex versus time (Figure 39). As hydrazine was adsorbed by kaolinite, the intensity of the non-intercalated d_{001} reflection at 12.36° 2θ (0.716 nm spacing) decreased, and a corresponding increase in intensity was observed for the "new" KH d_{001} reflection at 8.60° 2θ (1.03 nm spacing). Assuming that the ratio of the KH reflection against the kaolinite d_{001} reflection is directly proportional to the fraction of kaolinite intercalated, these data (Figure 39) indicated that after two hours of exposure to hydrazine more than 90 percent of the kaolinite was intercalated by hydrazine. The sharp decrease in intensity of the non-intercalated d_{001} reflection (Figure 39), however, may indicate that the intercalation process reached completion after only 30 minutes. X-ray diffraction (XRD) patterns showing the d_{001} reflections of kaolinite and of the kaolinite-hydrazine (KH) intercalate are presented in Figure 40. Upon expansion of the kaolinite lattice by hydrazine, the interlamellar spacing increased from 0.716 nm to 1.030 nm, an increase of 0.314 nm. This increase indicated that one molecular layer of hydrazine was adsorbed between each structural 1:1 layer of the clay lattice. By comparison to the other d_{001} reflections, the well-defined, relatively sharp 8.60° 2θ reflection in the XRD pattern of the KH complex obtained at 1 atmosphere of pressure (labeled "K:H air" in Figure 40), indicated that the KH complex was fairly well ordered under these conditions and that very little nonintercalated kaolinite remained after intercalation. These XRD results are in good agreement with those of Ledoux and White (Reference 26) who reported a d_{001} spacing of 1.040 nm for the KH complex at 25° C and 1 atm. Space filled drawings of the expanded kaolinite crystal structure (d_{001} = 1.030 nm) and of hydrazine (Figure 41) illustrate the approximate amount of space available for the intercalate in the interlamellar region and the corresponding dimensions of the hydrazine molecule.

Raman spectra in the 3600 to 3725 cm^{-1} region of kaolinite (top), and of a KH complex at 760 torr (bottom), are shown in Figure 42. The

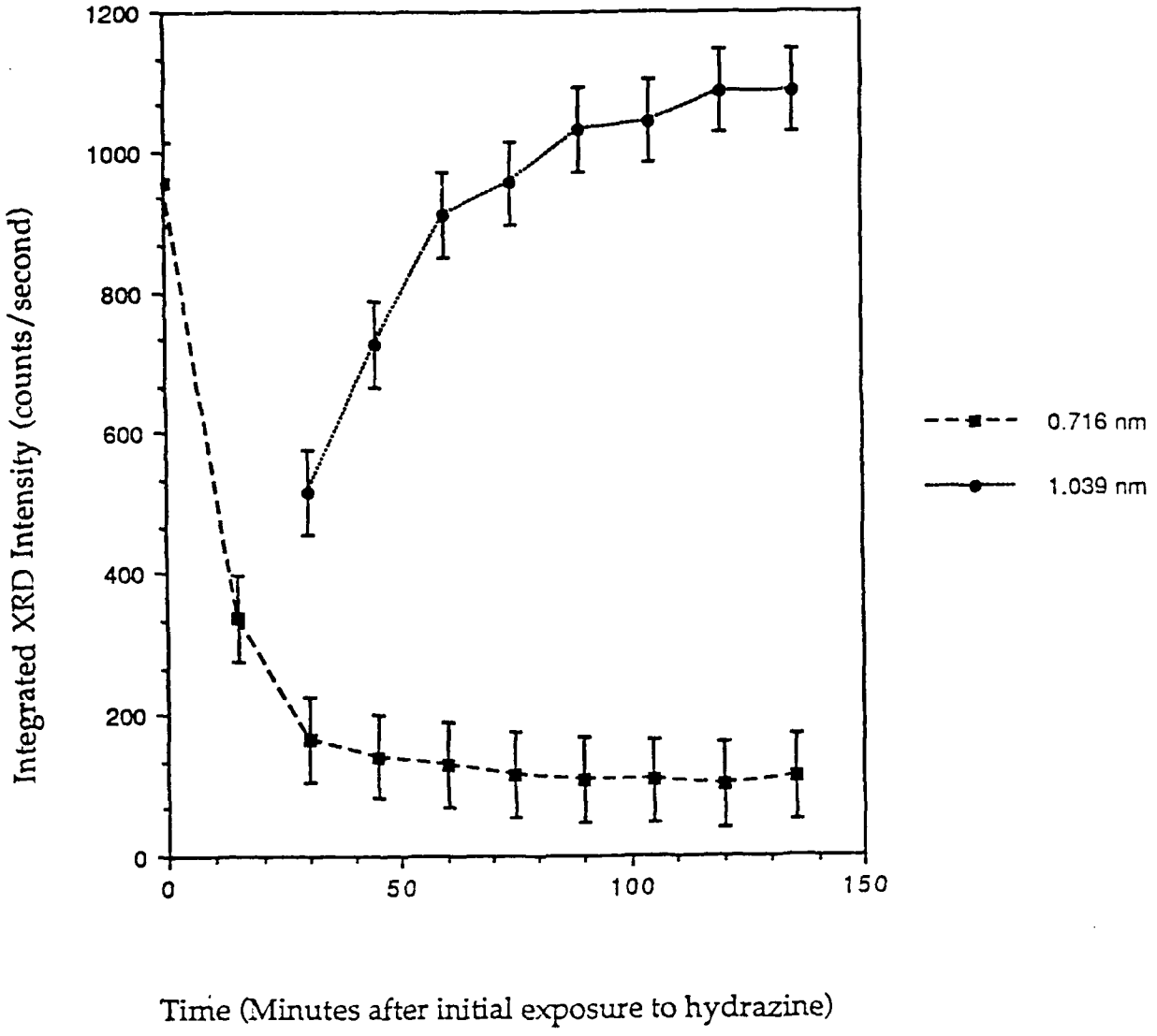


Figure 39. Intensity of the XRD d[001] Reflections Versus Time of Kaolinite (0.716 nm) and of the Kaolinite-Hydrazine Complex (1.039).

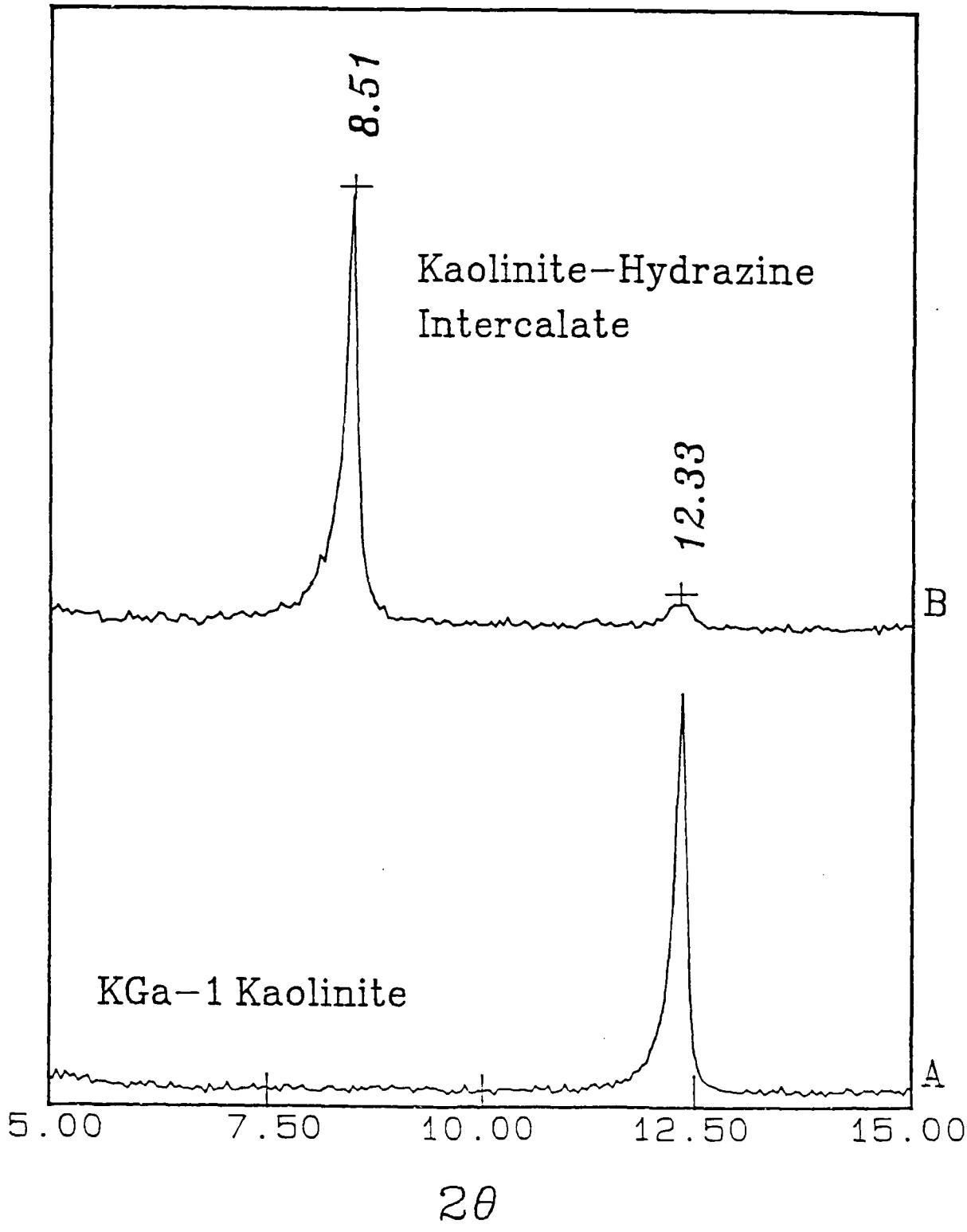


Figure 40. X-ray Diffraction Patterns of nonintercalated Kaolinite (top) and of the Kaolinite-Hydrazine Intercalation Complex at 760 Torr (bottom).

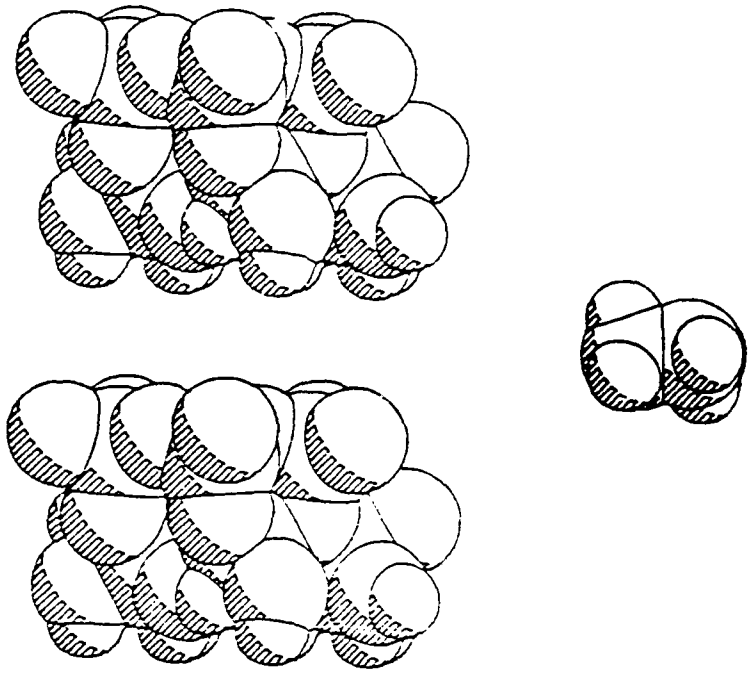


Figure 41. [010] Projection of the Expanded Kaolinite Structure after Expansion of Lattic by Hydrazine to a 1.03 nm d001 Spacing. A Hydrazine Molecule Drawn to the Same Scale Using van der Waals Radii is Shown on the Right.

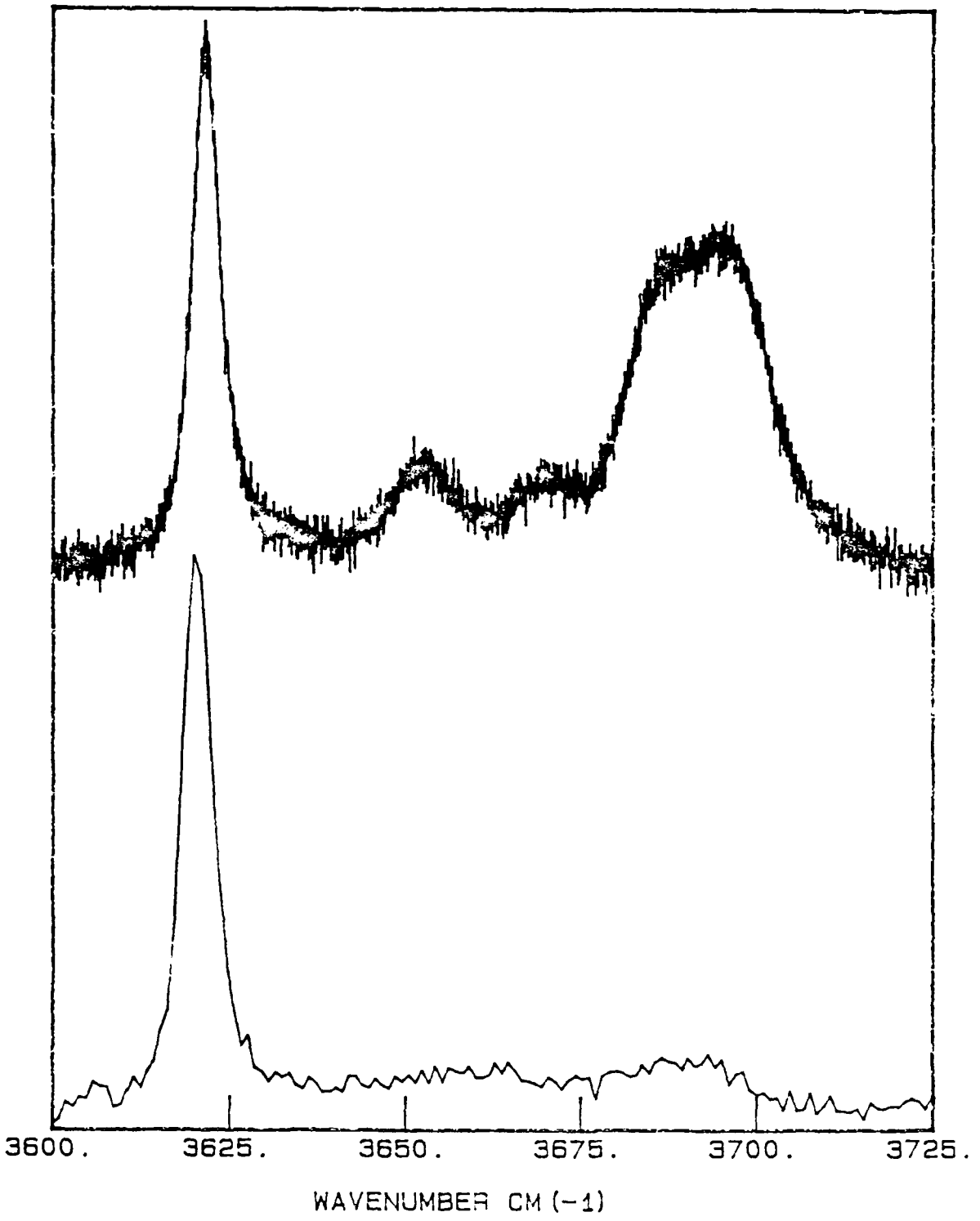


Figure 42. Raman Spectra in the 3600 to 3735 cm^{-1} Region of KH Complex at 760 Torr (bottom), and a-of a Dry Nonintercalated Mesa Alta Kaolinite Sample (top).

3620 cm^{-1} band was not perturbed upon intercalation under these conditions. In contrast, the intensities of the hydroxyl stretching bands at 3652, 3668, and 3694 cm^{-1} bands of kaolinite (Figure 43a) were reduced in intensity upon intercalation (Figure 43b-c), and the 3620 cm^{-1} band was not affected. These results are in agreement with a previous IR study of the KH complex (Reference 26).

The molecular structure of kaolinite projected along the [100] face (looking along the x-axis with the z-axis pointing up) is shown in Figure 32. The ball-and-stick drawing shown on the left illustrates the two distinct types of hydroxyl groups that reside within the crystal structure of kaolinite: the inner hydroxyl "sandwiched" between the Al-octahedral and Sitetrahedral layers of the clay lattice, and the inner-surface hydroxyl groups located on the basal plane of the Al-octahedral layer. Numerous infrared studies of kaolinite (References 10, 18-24) are in agreement that the 3620 cm^{-1} band is assigned to the inner hydroxyl group (Figure 32), and that this hydroxyl group has a considerably greater resistance to isotopic exchange with deuterium and to dehydroxylation at elevated temperatures in comparison to the labile inner-surface hydroxyl bands at 3652, 3669, and 3690-5 cm^{-1}.

FT-IR spectra of the KH complex in the 3550 to 3750 cm^{-1} region are shown in Figure 44 at 1 atm of pressure and at several reduced pressure values. As the KH complex is exposed to a reduced pressure, a new, higher-frequency band appeared at 3628 cm^{-1} and increased in intensity at the expense of the 3620 cm^{-1} band. Insofar as these authors are aware, a similar shift in frequency of the inner-hydroxyl stretching band induced by a guest intercalate has not been reported previously in the literature. There is little doubt because of the strong intensity borrowing between the two bands. The presence of two discrete inner-hydroxyl stretching bands suggests strongly that a different structural conformation of the intercalation complex was induced upon decreasing the pressure. A similar result was not observed in the Raman spectra of the KH complex because the sample was studied at 1 atm of pressure.

To confirm this hypothesis, XRD patterns of the evacuated complex were obtained at reduced pressure values (Figure 45). The d_{001} reflection of the KH intercalation complex was observed to <u>increase</u> from its

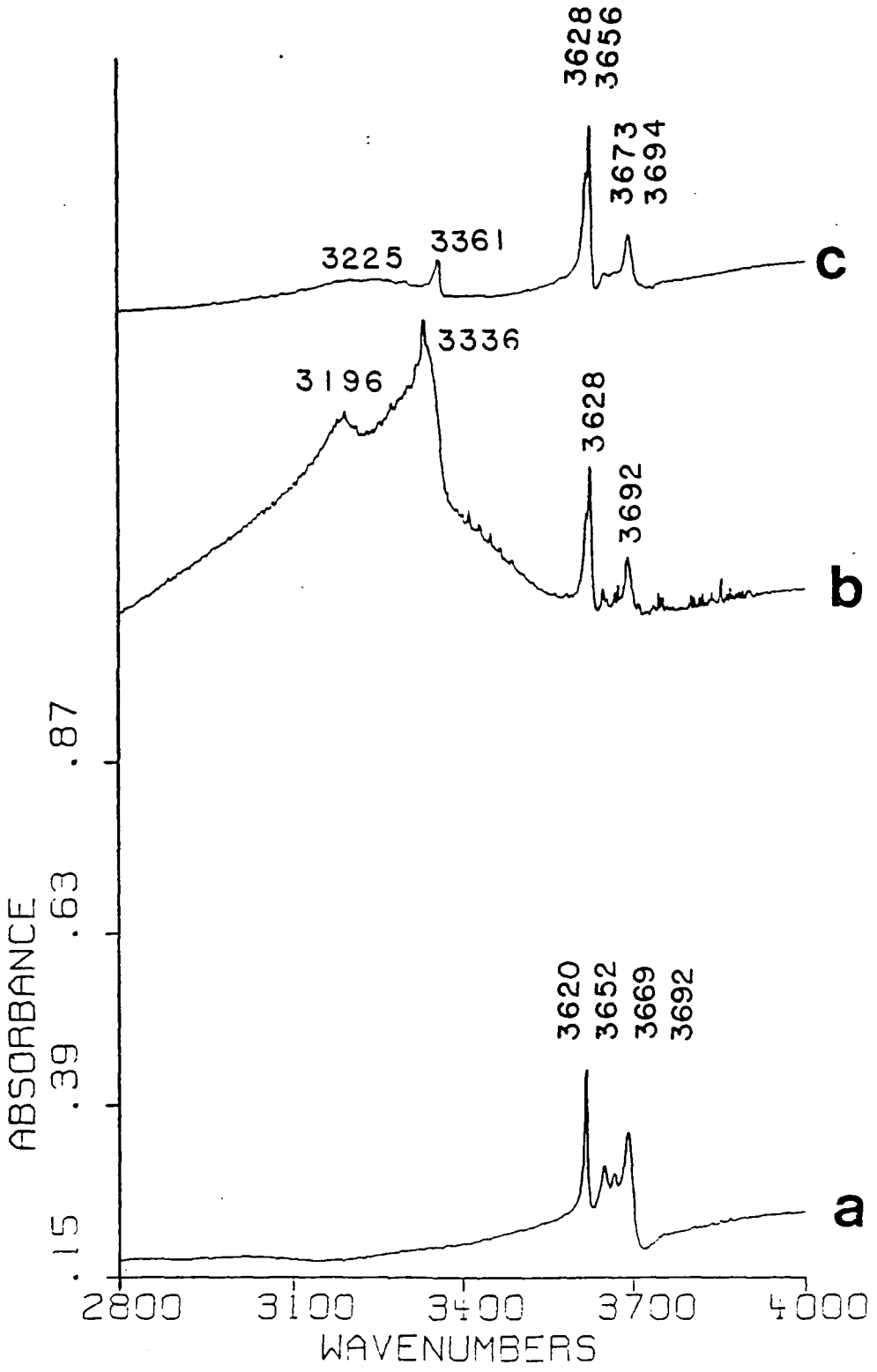


Figure 43. FT-IR Absorbance Spectra in the 2800 to 4000 cm^{-1} Region of Kaolinite (a), KH Complex at 760 Torr (b), and of the KH Complex Under a Vacuum of 0.0001 Torr (c).

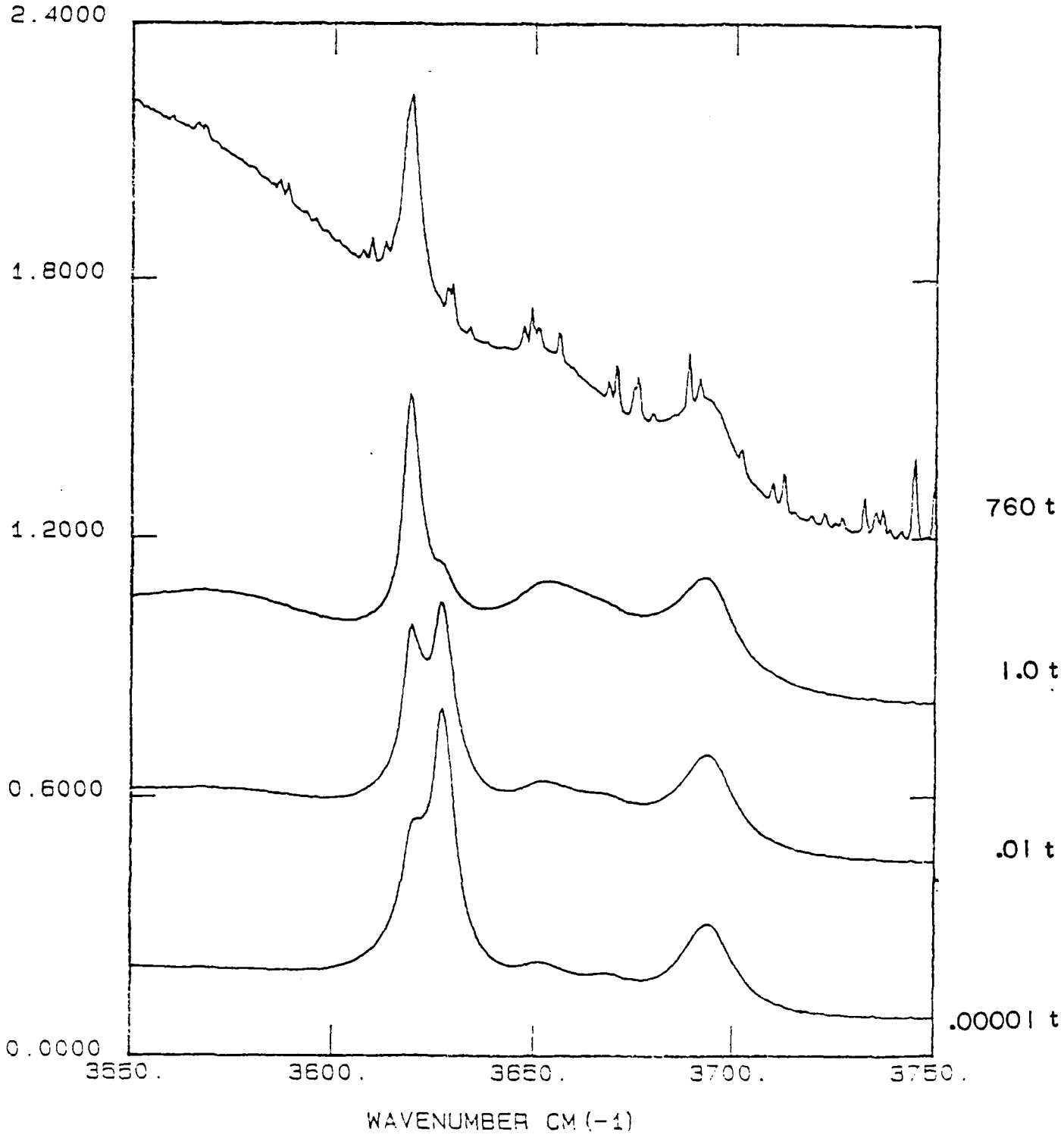


Figure 44. FT-IR Absorbance Spectra of the KH Complex at Pressure Values of 0.00001 Torr, 0.01 Torr, 1.0 Torr, and 760 Torr.

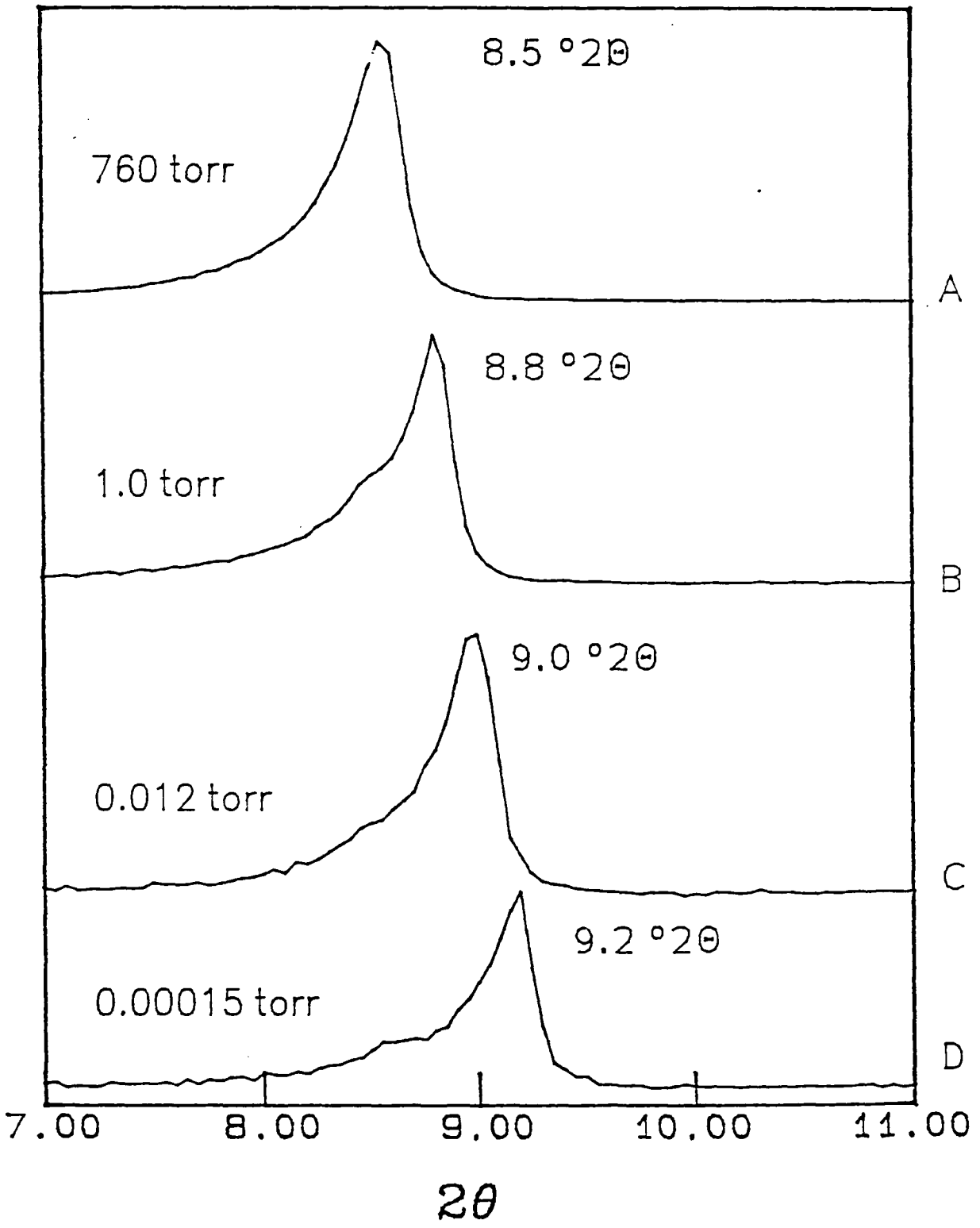


Figure 45. X-ray Diffraction Patterns Showing the d[001] Reflection of the Kaolinite-Hydrazine Complex at 760 Torr, 0.001 Torr, and 0.00001 Torr.

values of 8.60°2θ at 1 atm to 9.2°2θ, which corresponded to a decrease in the interlamellar spacing from 1.030 nm (760 torr) to 0.960 nm (0.00001 torr). Thus, the interlamellar spacing of the intercalation complex decreased by 0.07 nm upon evacuation, which decreased the available space for the guest species from 0.314 nm to .244 nm. This observed decrease in the interlamellar spacing of the KH complex provided conclusive evidence that a structural change of the KH complex did occur upon evacuation. This change was also reflected by the novel blue-shift of the inner-hydroxyl stretching band from 3620 to 3628 cm^{-1} which indicated that the $-NH_2$ moiety of the guest hydrazine species was brought into close contact with the inner-hydroxyl group after evacuation.

As the [100] ball and stick projection of kaolinite illustrates (Figure 32), the inner-hydroxyl group resides between the silica tetrahedral and aluminum octahedral layers and is not accessible by most intercalated species in the interlammellar region. The fact that hydrazine perturbed the stretching frequency of the inner hydroxyl group suggested strongly that the $-NH_2$ moiety keyed into the kaolinite surface resulting in a slight electrastatic repulsion between the $-NH_2$ and -OH groups responsible for the 8 cm^{-1} blue shift. Considering the molecular structure of kaolinite, the inner-hydroxyl group can only be approached by a guest molecule small enough to penetrate through the siloxane ditrigonal cavity. A space-filled [001] projection of the kaolinite structure showing the siloxane ditrigonal cavity and the molecular structure of hydrazine drawn to the same scale (Figure 46) illustrates qualitatively that the $-NH_2$ moiety is small enough to "fit" into the cavity.

J. CONCLUSION

In conclusion, the Raman and FT-IR spectra of the KH complex in the hydroxyl stretching region are in agreement in that both methods show a strong reduction in intensity of the inner-surface hydroxyl groups upon intercalation resulting from the formation of hydrogen bonds between the inner-surface hydroxyl groups of kaolinite and the interamellar hydrazine species. XRD patterns of the KH complex indicated that the interlamellar region increased in size by 0.314 nm at 1 atm of pressure to accommodate

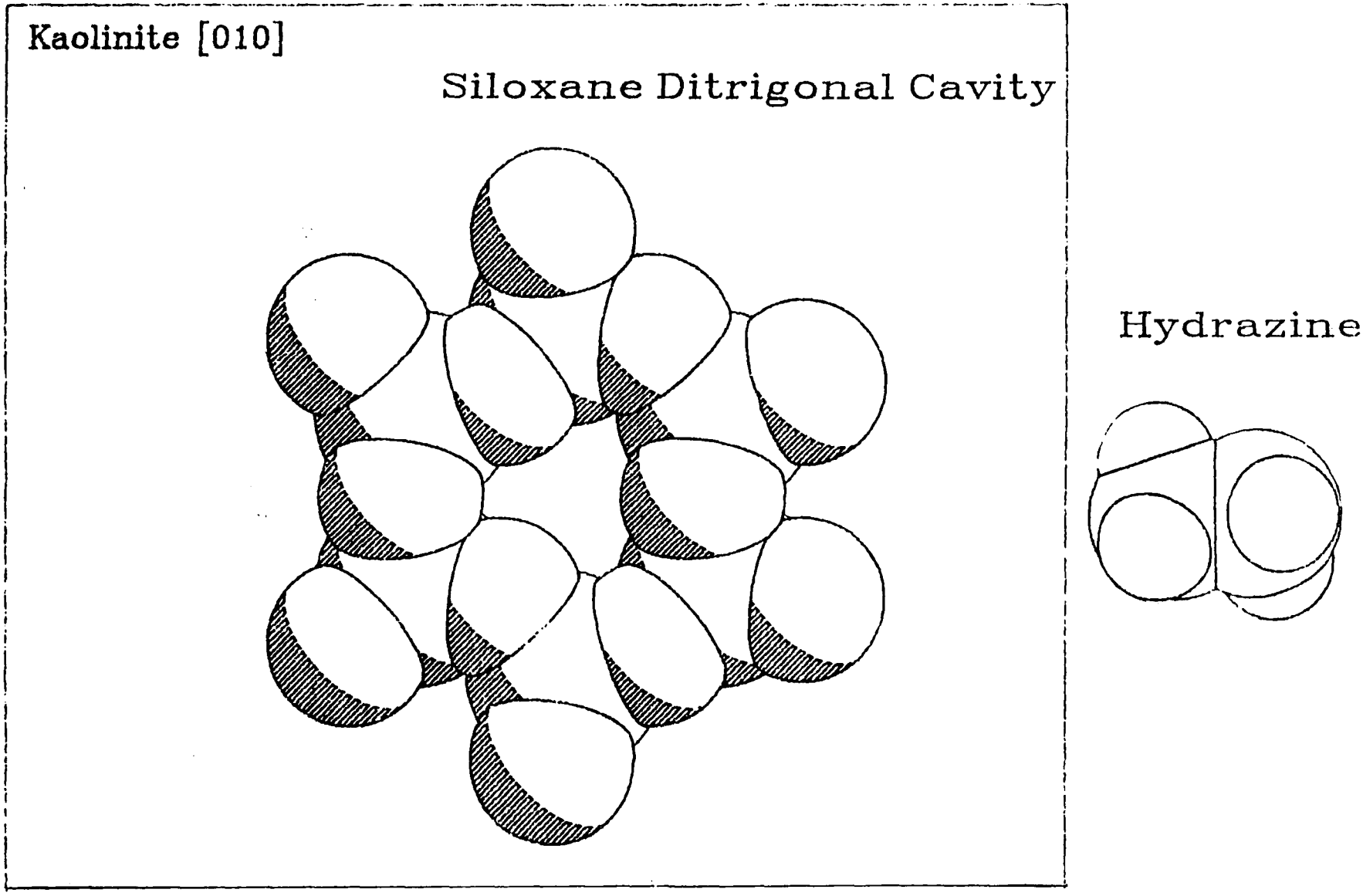


Figure 46. [001] Projection of the Kaolinite Siloxane Ditrigonal Cavity. A Hydrazine Molecule Drawn to Scale Using van der Waals is Shown on the Right.

the guest intercalate. As the pressure was reduced, however, this value decreased to 0.244 nm and the FT-IR spectra of the reduced pressure KH complexes showed clearly the presence of a new, higher frequency inner-hydroxyl stretching band at 3628 cm^{-1}. These results indicated that a structural change of the KH complex occurred at a reduced pressure, and that the $-NH_2$ moiety of hydrazine was brought into close contact with the innerhydroxyl group through the siloxane ditrigonal cavity. In addition, it was shown that the Raman-and IR-active hydroxyl stretching modes of kaolinite were sensitive probes of the interaction between hydrazine molecules and the kaolinite surface.

SECTION IV

THE SOIL MICROBIOLOGY OF HYDRAZINE AND MONOMETHYLHYDRAZINE

A. INTRODUCTION

Hydrazine, MMH and UDMH are toxic to many forms of bacteria. The activities of the autotrophic nitrifiers Nitrosomonas and Nitrobacter, dentrifying bacteria, and anaerobic methanogens were inhibited by these chemicals (Reference 27). In addition, the three chemicals prolonged the lag phase of growth as well as inhibited growth of the soil bacterium Enterobacter cloacae (References 28-30). Kane and Williamson (Reference 27) reported that among the three hydrazines, MMH was the most toxic to bacteria.

Despite hydrazine toxicity to bacteria, Kane and Williamson (Reference 27) demonstrated that hydrazine in small quantities was cometabolically degraded to nitrogen gas by Nirobacter. An enzyme system of nitrogen-fixing heterotrophic bacteria was able to convert hydrazine to ammonia (Reference 31). However, it was not clear that the bacteria could utilize hydrazine as a sole source of nitrogen for growth.

Hydrazine when present in natural waters (river, lake and pond) in small quantities, was not stable (Reference 4).

Information concerning the degradation of hydrazine and MMH and the effect of the chemicals on microbial activity in soils is not available. Accidental spills to soil can occur during transportation and storage. Therefore, this study was initiated to determine degradation rates of hydrazine and MMH in soil and their effect on soil microbial activity. In addition, we isolated bacteria capable of degrading hydrazine and MMH from soil, and used the isolates to enhance degradation in soil and water samples contaminated with hydrazine or MMH.

B. MATERIALS AND METHODS

Hydrazine sulfate, hydrazine monohydrate and MMH were purchased from Aldrich Chemical Company (Milwaukee, WI). Hydrazine sulfate was used in all experiments unless otherwise specified. Uniformly labeled

^{15}N-hydrazine sulfate was obtained from Icon (Summit, NJ), with the chemical consisting of 98 atom percent of ^{15}N.

Arredondo fine sand (Grossarenic Paleudult) was used for this study. The soil, which had never been exposed to hydrazine and MMH, was air-dried and sieved to pass a 2-mm sieve. The sample had a pH of 5.7, and 1.7 percent of organic carbon.

Two hundred grams of soil (oven-dry weight basis) were placed in 500 mL glass bottles or flasks, and 16 mL of distilled water were added. Appropriate amounts of hydrazine sulfate were added to give hydrazine concentrations of 0, 2.5, 10, 25, 100, 125, 250, and 500 µg/g, or MMH concentrations of 0, 10, 50, 100, and 500 µg/g. After mixing, the bottles were weighed and incubated at 25°C. For determination of hydrazine and MMH residues, as well as bacterial and fungal populations in the soil, 10 grams of soil samples were removed. For determination of nitrate and ammonia, 15 grams of samples were removed. Once a week the weights of the soil samples were checked, and distilled water was added to compensate any water loss.

Hydrazine was determined by the colorimetric method of Watt and Chrisp (Reference 32). Plastic centrifuge tubes containing 10 grams soil samples were deoxygenated by flushing with a stream of N_2, and the samples were extracted three times with 20 mL of deoxygenated 0.1M NaCl. 0.1 to 1 mL aliquots were transferred to 10 mL of the color-developing agent, p-dimethylaminobenzaldehyde, and the resulting mixtures were diluted to 25 mL by adding 1 M HCl. Specific-ion electrodes were employed to determine NH_4^+ and NO_3^- in the soil. After mixing with 0.1 g of calcium sulfate, 15 grams soil samples were extracted with 45 mL of distilled water. Twenty and 10 mL aliquots were used for determination of NO_3^- and NH_4^+, respectively.

MMH was determined by the colorimetric method of Reynolds and Thomas (Reference 33). Briefly, 10 grams of soil were placed in plastic centrifuge tubes and extracted three times with 20 mL of cold and deoxgenated 0.1M HCl. 0.01 to 1.0 mL of the extracts were transferred to 4 mL of 10% trichloroacetic acid and adequate amounts of deionized water were added to give a final volume of 5 mL. Four mL of each mixture were mixed with 5 mL of the color-reagent, p-dimethylaminobenzaldehyde. After 30 minutes, optical densities of the resultant mixtures were read at 485 nm.

In addition, uniformly-labeled ^{14}C-MMH, along with unlabeled MMH, were also used for determination of disappearance and mineralization of MMH in soil. ^{14}C-MMH was purchased from Amersham Corporation (Arlington Heights, IL) and had a specific activity of 6 µCi/mmol and 98% radiopurity. $^{14}CO_2$ trapped in KOH and ^{14}C-activity in the HCl extracts were determined by scintillation counting. ^{14}C-activity remaining in the extracted soil was determined by combusting to $^{14}CO_2$ in a Packard Tri-Carb sample oxidizer as described previously by Ou et al. (Reference 34). The evolved $^{14}CO_2$ was trapped in a scintillation cocktail containing an organic amine, and counted by scintillation counting. All soil samples had been incubated at 25°C.

Dilution-plate count methods as described by Ou et al. (Reference 35) were used to determine bacterial and fungal populations in the soil samples. Carbon dioxide which evolved from the samples in closed glass bottles was trapped in KOH and determined by titration (Reference 36). All experiments were duplicated, with the exception of the soil respiration experiment, which was done in triplicate.

Basal mineral medium consisted of (per liter of distilled water) 4.8 g of K_2HPO_4, 1.2 g of KH_2PO_4, 1 g of NH_4NO_3, 0.25 g of $MgSO_4 \cdot 7H_2O$, 0.04 g of $CaCl_2$, and 0.001 g of $Fe_2(SO_4)_3$. Glucose, 0.2 g/mL, and hydrazine sulfate, 0.01 g/mL, were sterilized separately by autoclaving and filtration, respectively. For maintenance of bacterial cultures, 10 mL of the glucose and 10 mL of the hydrazine were added to 1 L of the basal mineral medium. Tryptic soy agar and broth (Difico) were also used for purification and maintenance of bacterial cultures.

One hundred grams of Arredondo fine sand (Grossarenic Paleudult) were repeatedly treated with 100 µg/g of hydrazine. This soil had previously been shown to readily degrade hydrazine at 100 µg/g (Reference 3) After four applications, 10 grams of the soil were transferred to a culture tube containing 10 mL sterile distilled water. After the tube was briefly shaken, 0.3 mL aliquots of the suspension were spread on NH_4NO_3-free basal mineral agar plates supplemented with glucose and hydrazine. The plates were incubated at 25 C° for observation of the development of bacterial and fungal colonies. Bacterial and fungal colonies of different appearance were further transferred to new plates. The bacterial isolates were also transferred to tryptic soy agar plates for purification.

Growth of bacterial cultures was determined turbidimetrically at 550 nm with a Spectronic 20 spectrophotometer.

Eighteen-h old cultures were harvested by centrifugation at 20,000xg for 15 min at 4°C, washed once with potassium phosphate buffer (K_2HPO_4, 4.8 g/L and KH_2PO_4, 1.2 g/L), and resuspended in the same buffer.

Frozen cells (3 g) were suspended in 6 mL of the potassium phosphate buffer. The suspension was irradiated with ultrasonic waves for 9 minutes with an ultrasonic cell disruptor (Sonified W140, Heat Systems-Ultrasonic), using a titanium probe at 20,000 Hz and 2 amperes. The resulting suspension was centrifuged at 30,000xg for 30 minutes. The clear supernatant was used as a source of crude cell extracts. Protein concentration in cell extracts was determined by the method of Bradford (Reference 37), using bovine serum albumin as the standard.

Hydrazine was colorimetrically determined using the method of Watt and Chrisp (Reference 32). 0.01-0.2 mL of supernatants from cell cultures, cell suspensions, or cell extracts was transferred to 25 mL of volumetric flasks which had contained 10 mL of color-developing agent, p-dimethylaminobenzaldehyde, and sufficient 1 M HCl was added to make a total volume of 25 mL. Optical densities of the samples were determined at 458 nm using a Spectronic 20 spectrophotometer.

Ammonia concentrations in cell suspensions were determined by a specific ammonia electrode as described previously by Ou and Street (Reference 3).

Nitrate concentrations in cell suspensions were determined by a nitrate-specific electrode. Nitrate concentrations in cell suspensions were determined colorimetrically at 520 nm, using the method of Barnes and Folkard (Reference 38).

$^{15}N_2$ gas, which was produced in cell suspensions treated with ^{15}N-hydrazine sulfate, was determined by mass spectometry. 6 ml aliquots of cell suspension were placed in 20 mL glass serum bottles which contained 0.6 or 0.9 mg of ^{15}N-hydrazine. 0.2 mL of air in the headspace was withdrawn for mass spectral analysis.

A strain of Achromobacter sp. and a strain of Pseudomonas sp., which had the capacity to degrade hydrazine in the presence of a second nitrogen source such as ammonium nitrate, were used for studing the

enhancement of hydrazine and MMH degradation in contaminated soil and water. Both bacteria were isolated from Arredondo fine sand. The bacteria were maintained, as described by Ou (Refernce 39) in a basal mineral medium containing 10 ug/g of MMH or 100 ug/g of hydrazine.

MMH in culture fluids and cell suspensions was determined by the colorimetric method of Reynolds and Thomas (Reference 33). In addition, ^{14}C-MMH was used for determination of MMH disappearance and formation of metabolites. ^{14}C-activity in culture fluids was determined by scintillation counting. ^{14}C-metabolites were determined by organic-solvent extraction, thin-layer chromatography (TLC)-autoradiography, and scintillation counting (Refernce 34). ^{14}C-activity in culture fluids was determined by extracting with an equal volume of chloroform and ethyl ether. The organic extracts, after removing moisture with anhydrous sodium sulfate, were concentrated to 1.5-2.0 mL in a stream of N_2 gas. Aliquots (10 to 20 µL) of the extracts were spotted on commercial silica gel G plates. The TLC developing-solvent system was hexane-chloroform-methanol (7:2:1, v/v). Radioactive areas on each plates were detected by placing Kodak SB-5 X-ray films on the plates. The radioactive areas on the plates were scraped, transferred to scintillation vials, and quantified by scintillation counting.

$^{14}CO_2$, which evolved from growing cultures in glass Erlenmeyer flasks containing ^{14}C-MMH, was trapped in small stainless steel vials containing 1 gram of KOH pellets. The vials were hung under the rubber stoppers in the flasks using stainless steel wire. The flasks were then tightly closed with the stoppers. After removing the KOH from the flasks it was diluted with water to 4 mL, and ^{14}C-activity in the KOH solutions was determined by scintillation counting.

Growth of bacterial cultures was determined turbidimetrically at 550 nm with a Spectron 20 spectrophotometer.

Sixteen- to 20-hour-old cultures were harvested by centrifugation in the cold, washed once with cold phosphate buffer (pH 7.2), and resuspended in the same buffer.

All bacterial cultures and suspensions were incubated at 25°C. All experiments were carried out in duplicate.

Two river water samples, two lake water samples, tap water, and distilled water were used in this study. Selected properties of the waters samples are shown in Table 15.

TABLE 15. SELECTED PROPERTIES OF THE SIX WATERS USED IN THIS STUDY.

Water	Cu (μg/mL)	Fe (μg/mL)	Bacteria (cfu/mL) X10^{-3}	Fungi (cfu/mL)	pH	Suspended solid (mg/mL)
Santa Fe River	0.04	0.04	206	4	7.7	3
Prairie Creek	0.01	0.24	1	3	6.6	3
Lake Alice	0.20	0	25	22	7.4	3
Newmans Lake	0	0.28	9	9	7.7	3
Tap water	0	0	0	0	8.5	3
Dist. Water	0	0	0	0	6.4	0

Achromobacter sp. was maintained in a basal mineral medium containing hydrazine sulfate and glucose as described by Ou (Reference 39). 30 mL of 18 hour-old bacterial culture was centrifuged at 20,000xg for 20 minutes. The cell pellets were suspended in 5 mL of water. After adding an appropriate amount of sterile aqueous hydrazine (0.04 - 0.2 mL), the samples were incubated at 25°C for 2 hours.

The aqueous hydrazine (250 or 2500 μg/mL) was sterilized by filtration through a 0.22 μm filter, and then stored in the dark at 4°C. Under this condition, the hydrazine solution was stable for more than a month. In addition, autoclaved bacterial cells were tested for their capacity to degrade hydrazine in autoclaved and filtered water samples. Hydrazine sulfate was used for all experiments, unless otherwise specified.

Water samples (5 mL) were buffered with phosphates (0.024 g of K_2HPO_4 and 0.024 g of KH_2PO_4). Eighteen hour-old bacterial cells were suspended in the buffered waters. After hydrazine had been added, the samples were incubated at 25°C for 2 hours, after which the cell suspensions were centrifuged at 20,000xg for 20 minutes. 0.01 to 0.1 mL of the supernatant solutions was used for determination of hydrazine concentration using the colorimetric method described earlier (Reference 32).

One mL of the 18 hour-old cell suspension of Achromobacter sp. (12±1 mg/mL) was also added to 10 grams of air-dry Arredondo soil in a plastic centrifuge tube, and an appropriate amount of sterile hydrazine was immediately added to give hydrazine concentrations of 25, 50, or 100 µg/g of soil. An identical experiment was set up for autoclaved soil. In addition, the bacterial cells (2.2±0.2 mg/mL) were suspended in 5 mL of either distilled water or buffered distilled water in plastic centrifuge tubes which contained 0.5 g of autoclaved or nonautoclaved Arredondo soil. Appropriate amounts of sterile hydrazine (25 and 50 µg/mL) were added to each suspension. All samples were incubated at 25°C for one hour. After incubation, the soil suspensions were immediately centrifuged in the cold (4°C) at 20,000xg for 15 minutes. Corresponding soil samples were immediately suspended in 30 mL of cold 0.1 M NaCl, and centrifuged in the cold. Hydrazine in each supernatant was determined colorimetrically as described earlier.

C. RESULTS AND DISCUSSION

1. Hydrazine

a. Degradation in Soil

At low concentrations, hydrazine disappeared rapidly from Arredondo soil. For example, at 10 µg/g, hydrazine disappeared completely in 1.5 hours. Autooxidation appeared to be the principal factor contributing to the disappearance of the chemical from soil, as less than 3 percent of the applied hydrazine was recovered from sterile soil. Even at 100 µg/g hydrazine disappeared completely in 1 day and, at 500 µg/g, the chemical disappeared completely in 8 days (Table 16). Biological degradation was a relatively minor factor, although hydrazine consistently disappeared from sterile soil at somewhat slower rates. By comparing the hydrazine loss from sterile and nonsterile soils, it appeared that biological degradation was responsible for about 20 percent of the disappearance.

Since hydrazine is partly degraded biologically in this soil, it is of interest to determine if hydrazine is metabolized to

TABLE 16. HYDRAZINE IN STERILE AND NONSTERILE ARREDONDON SOIL.

Days	Hydrazine (percent remaining)			
	100 µg/g		500 µg/g	
	Sterile	Nonsterile	Sterile	Nonsterile
0.05	83±11	62±3	97±3	93±2
1	8±0	0	71±1	62±2
2	0	0	52±0	39±3
3	0	0	39±2	25±4
6	0	0	13±1	3±1
8	0	0	8±1	0

ammonia, which can serve as a nitrogen source for growth. However, we found no evidence of hydrazine being converted to ammonia. The levels of NH_4^+ in Arredondo soil, both with and without treatment with hydrazine at 100 µg/g, were essentially the same following 7 days of incubation.

b. Microbial Degradation

Fungal colonies were found to develop in the NH_4NO_3-free basal mineral agar plates supplemented with glucose and hydrazine. However, they either did not grow or grew poorly in liquid basal mineral medium supplemented with hydrazine and glucose with or without NH_4NO_3. The fungi were found to have little capacity to degrade hydrazine. Hence, they were not used for further investigation. Bacterial colonies of small size developed a few days after fungal colonies appeared. The bacteria all failed to grow in liquid NH_4NO_3-free basal mineral medium supplemented with glucose and hydrazine. However, some of the bacteria grew when supplemented with NH_4NO_3. An Achromobacter sp., a Bacillus sp., and a Pseudomonas sp. were found to have capacity to degrade hydrazine. Both Bacillus sp. and Pseudomonas sp., when grown in glucose-amended basal mineral medium containing hydrazine, had a lag growth period of 3 to 5 days. The cell suspensions of the two bacteria could only degrade hydrazine at concentrations of 25 µg/ml and lower. The Achromobacter sp. not only had a short lag growth period but also

degraded hydrazine at concentrations greater than 100 µg/ml. Therefore, this bacterium was chosen for further study.

The Achromobacter sp. in glucose-amended basal mineral medium containing 25 and 51 µg/mL hydrazine exhibited 4 and 8 hours of lag growth period, respectively (Figure 47). The bacterial culture at both hydrazine concentrations reached maximal growth by 48 hours. Unlike the bacterial growth, hydrazine was degraded without a lag period. Degradation had levelled off before cell growth reached maximized.

Hydrazine at 51 µg/mL in the glucose-amended basal mineral medium in the absence of the bacterial culture was much more stable than in the presence of the culture (Figure 47). After 48 hours of incubation 97 percent of hydrazine remained in the culture-free medium, whereas only 4 percent of the chemical remained in the culture medium during the same incubation period. In addition, hydrazine at 25 µg/mL was also stable in the culture-free glucose-amended basal mineral medium. Hydrazine was also found to be stable in the basal mineral medium, in a glucose solution (0.2 g/mL), and in distilled water.

Hydrazine, at various concentrations ranging from 25 to 162 µg/ml, was used for testing the degradative capacity of cell suspensions of the Achromobacter sp. which had been grown in hydrazine and glucose-amended basal mineral medium. Table 17 shows that hydrazine at initial concentrations of 25, 50, and 90 µg/mL was degraded to near completion within 2 hours, and at 120 and 162 µg/mL more than 50 percent of the chemical was degraded. Hydrazine in heated and autoclaved suspensions was stable and, in fact, no hydrazine in the autoclaved suspension was degraded during 2 hours of incubation.

Nitrate, nitrite and ammonia were not formed in cell suspensions treated with hydrazine. $^{15}N_2$ gas was detected in air samples from the headspace of serum bottles containing the bacterial cell suspension and ^{15}N-hydrazine. ^{15}N-hydrazine was not detected in the air samples.

The cell suspension of the Achromobacter sp. which had been grown in the basal mineral medium without hydrazine also had a capacity to degrade hydrazine (Table 18). The bacterium grown in the basal mineral medium with KNO_3 or $(NH_4)_2SO_4$ as a sole source of nitrogen also degraded hydrazine. Furthermore, bacterial cells grown in rich complex

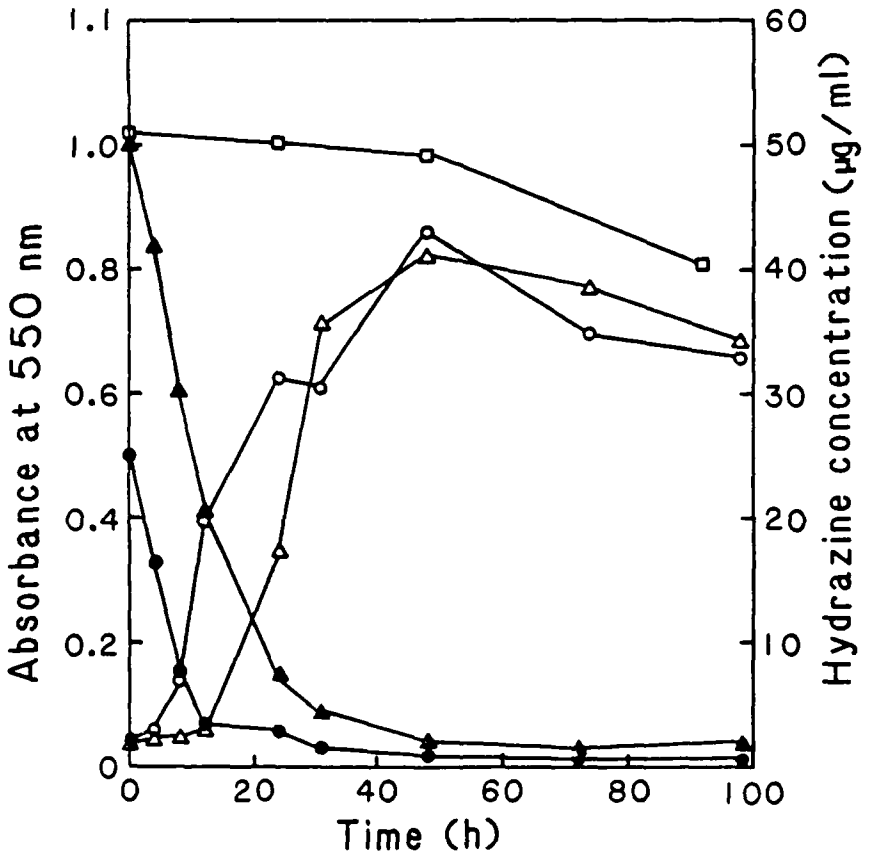


Figure 47. Hydrazine degradation and growth of the Achromobacter sp. Legend: ○ and △, absorbance of culture fluids with initial hydrazine concentrations of 25 and 51 µg/mL, respectively; ● and ▲, hydrazine concentrations in culture fluid; and □, hydrazine concentrations in the culture-free glucose-amended medium.

medium such as tryptic soy broth also had a capacity to degrade hydrazine.

The Achromobacter sp. also degraded the nonsalt form of hydrazine. It was found that only 6 percent of applied hydrazine monohydrate (25 µg/mL) remained in the cell suspension after 2 hours of incubation. Autoclaved cells lost their capacity to degrade hydrazine monohydrate.

Crude cell extracts of the Achromobacter sp. also had a capacity to degrade hydrazine. The specific activity of the extracts toward the reduction of hdyrazine was estimated to be 0.30 µmol/h/mg

TABLE 17. HYDRAZINE IN CELL SUSPENSIONS OF THE ACHROMOBACTER SP. GROWN IN BASAL MINERAL MEDIUM CONTAING HYDRAZINE[a]

Cell suspension	Initial N_2H_4 (µg/mL)	Percent Reduction after 2 hours
Live[b]	25	96
Live[b]	50	95
Live[b]	90	94
Live[b]	104	84
Live[b]	120	56
Live[b]	162	52
Heated[c]	50	4
Autoclaved[d]	50	0

[a] Cell suspensions were incubated at 25°C for 2 hours.

[b] Dry cell weight of the suspension was 3.8 mg/mL.

[c] Cell suspension was heated in a waterbath at 90°C for 10 minutes.

[d] Cell suspension was autoclaved at 121°C for 15 minutes.

TABLE 18. DEGRADATION OF HYDRAZINE IN CELL SUSPENSIONS OF THE ACHROMOBACTER SP. GROWN IN MEDIA WITHOUT HYDRAZINE[a]

Cell suspension	Initial N_2H_4 (µg/mL)	Percent Reduction after 2 hours
Grown in glucose - amended basal mineral medium[b]	42	94
Grown in glucose - amended basal in mineral medium[b]	83	86
Grown in glucose - amended basal mineral medium[c]	27	93
Grown in glucose- amended basal mineral medium[d]	27	97
Grown in tryptic soy broth[e]	50	86
Grown in tryptic soy broth[e]	125	83

[a] Cell suspensions were incubated at 25°C for 2 hours.

[b] Cell dry weight 3.7 mg/mL. Nitrogen source in the basal mineral medium was NH_4NO_3.

[c] Cell dry weight 3.6 mg/mL. Nitrogen source in the basal mineral was KNO_3.

[d] Cell dry weight 3.8 mg/mL. Nitrogen source in the basal mineral was $(NH_4)_2SO_4$.

[e] Cell dry weight 4.0 mg/mL.

protein. Heated extracts lost their capacity to degrade hydrazine. Hydrazine in phosphate buffer was not degraded.

These results suggest that the enzyme system for the metabolism of hydrazine by the Achromobacter sp. is constitutive. This is evidenced by the fact that hydrazine is rapidly degraded without a lag. The failure of autoclaved and heated cells, and of heated cell extracts, to degrade the chemical indicates that degradation during short incubation is principally microbial. For long incubations, autooxidation may play a role as well (Reference 40). Under the systems used in this study, the role of autooxidation is negligible.

Similar to the growth response of the soil bacterium Enterobacter cloacae to hydrazine (Reference 28, 30), an increase in hydrazine concentration caused an extension of the lag growth period for Achromobacter sp. However, lag growth periods were short. This is because more than 50 percent of the hydrazine has already been degraded at the onset of exponential growth. Obviously the enzymes responsible for degradation of the chemical are not poisoned or deactivated by the chemical.

Inability of the Achromobacter sp. to grow on hydrazine as a sole source of nitrogen indicates that the metabolic process for hydrazine is cometabolic. The Achromobacter sp. utilizes nitrate or ammonia as a sole source of nitrogen for growth, and these cells degrade hydrazine. Thus, it is likely that some common enzymes responsible for metabolism of nitrate and ammonia also have a capacity to degrade hydrazine. Oxidation of hydrazine to nitrogen gas by Nitrosomonas sp. has also been suggested to be a cometabolic process (Reference 27).

It is understandable that nitrate, nitrite and ammonia were not the degradation products of hydrazine. If those products had been formed Achromobacter sp. should be able to utilize hydrazine for growth. Likewise, it is also unlikely that hydroxylamine is a degradation product. Hydroxylamine is toxic and unstable (References 41, 42), and is suspected to be an intermediate of nitrate metabolism.

The observation that some common heterotrophic soil bacteria, such as Achromobacter sp., Bacillus sp. and Pseudomonas sp., as well as autotrophic nitrifiers reported by Kane and Williamson (Reference 27), can degrade hydrazine suggests that microbial degradation, in addition to

autooxidation, may play a role in removing this chemical from the environment. In light of the rapid degradation by Achromobacter sp., which was observed at concentrations beyond 100 µg/mL, the bacterium may have potential for use in the detoxication of hydrazine from contaminated soils and waters, and from wastes.

c. Enhancement of Degradation

Hydrazine was not degraded in any of the six waters during 2 hours of incubation. Hydrazine was not stable, however, in these waters when suspended with 18 hour-old cells of Achromobacter sp. The degree of stability in the cell suspensions depended upon hydrazine concentration and type of water. Table 19 shows percent reduction of hydrazine in six waters suspended with the Achromobacter sp. after 2 hours of incubation. The initial hydrazine concentrations ranged from 11 to 75 µg/mL. Bacterial cells suspended in Santa Fe River water were the most active in degrading hydrazine. Bacterial cells suspended in Lake Alice water also had a high capacity to degrade the chemical. It appeared that the Achromobacter sp. in Santa Fe River water and Lake Alice water declined in their capacity to degrade hydrazine at 50 µg/mL. The same bacterium in other waters started to decline in degradation capacity at 25 µg/mL. No degradation was observed in the six waters when suspended with autoclaved bacterial cells during two hours of incubation. Similar degradation capacity was observed whether the bacterial cells were suspended in autoclaved or filtered water samples.

When the water samples were buffered with phosphates, the degradation capacity of the Achromobacter sp. in Santa Fe River water, Lake Alice water, tap water, and distilled water was enhanced (Table 20). Even in distilled water 57 and 53 percent of the hydrazine was degraded when initial hydrazine concentrations were 75 and 100 µg/mL, respectively. pH values for the buffered waters were at least 3.6 units higher than for unbuffered waters, with the exception of the Santa Fe River water. This water appeared to have a high buffer capacity against hydrazine sulfate (Table 21). The critical pH value was approximately 3.5. Below this range the capacity for the organism to degrade hydrazine decreased drastically.

TABLE 19. HYDRAZINE DEGRADATION IN WATERS SUSPENDED WITH A SOIL BACTERIUM ACHROMOBACTER SP.[a]

Water	Initial N_2H_4 (µg/mL)	Percent Reduction after 2 hours	Initial N_2H_4 (µg/mL)	Percent Reduction after 2 hours
Santa Fe River	11	90	25	92
	50	96	75	51
Lake Alice	11	91	25	96
	50	78	75	46
Newmans Lake	11	90	25	72
	50	30		
Prairie Creek	11	90	25	76
	50	28		
Tap Water	11	91	25	80
	50	52	75	9
Distilled water	11	90	25	32
	50	22	75	5

[a]Bacterial cells were added at 2.2 ± 0.2 mg/mL.

Table 22 shows the stability of hydrazine in autoclaved and nonautoclaved waters. Among the six unsterile waters, hydrazine in Santa Fe River water was the most unstable. Hydrazine in this water disappeared completely in 14 days. Even in autoclaved water, 66 percent of the hydrazine was degraded in 14 days. On the other hand, hydrazine in Newmans Lake water and in Prairie Creek water was very stable, even more so than in distilled water. No degradation occurred in the two waters during 14 days of incubation. By comparing the degradation results in sterile and nonsterile waters, it appeared that biological degradation was more important initially for the Santa Fe River water, whereas autooxidation was more important initially for the Lake Alice water. It was not clear why Newmans Lake water and Prairie Creek water did not degrade hydrazine, and why hydrazine in these waters was more stable than in distilled water. Bacterial and fungal populations in these two waters were very low (see Table 15). No bacteria or fungi were detected in the distilled water sample.

TABLE 20. HYDRAZINE DEGRADATION IN BUFFERED WATERS SUSPENDED WITH CELLS OF AN ACHROMOBACTER SP.[a]

Water	Initial N_2H_4 (µg/mL)	Percent Reduction after 2 hours	Initial N_2H_4 (µg/mL)	Percent Reduction after 2 hours
Santa Fe River	75	63	100	49
Lake Alice	75	63	100	52
Tap water	75	65	100	53
Distilled water	75	57	100	53

[a]Bacterial cells were added at 2.2±0.2 mg/mL.

TABLE 21. pH VALUES OF WATER AMENDED WITH VARIOUS CONCENTRATIONS OF HYDRAZINE

Water	pH				
	Hydrazine concentration (µg/mL)				
	25	50	75	50	75
	Unbuffered			Buffered	
Santa Fe River	6.8	6.4	5.9	6.7	6.7
Prairie Creek	3.4	3.2	3.0	6.7	6.7
Lake Alice	6.4	5.5	3.4	6.7	6.7
Newmans Lake	3.5	3.2	3.0	6.7	6.7
Tap water	5.8	3.5	3.1	6.7	6.7
Distilled water	3.3	3.1	2.9	6.7	6.7

TABLE 22. HYDRAZINE DEGRADATION IN NONSTERILE AND STERILE WATER.[a]

Time (days)	Percent Reduction					
	Santa Fe River	Lake Alice	Newmans Lake	Prairie Creek	Tap water	Dist. water
1	28[b](5)[c]	0(8)	0(0)	0(0)	2(0)	0(0)
2	40(9)	9(9)	0(0)	0(0)	2(0)	0(0)
5	85(33)	42(34)	0(0)	0(0)	12(0)	0(0)
7	96(45)	56(48)	0(0)	0(0)	19(2)	2(0)
14	100(66)	86(71)	0(0)	0(0)	28(8)	2(0)

[a] Initial hydrazine concentration was 25 µg/mL

[b] Unsterile water

[c] Sterile water

Although cupric ions have a capacity to catalyze the auto-oxidation of hydrazine in waters (Reference 40), no relationship between copper content and rate of hydrazine degradation was found for the waters used in this study. Likewise, ferric ions may also catalize oxidation of the chemical in waters. Again no correlation was found. However, the levels of copper and iron in the waters were either very low or negligible (Table 15). At these low levels cupric ions or ferric ions may not exert any catalytic effect on hydrazine oxidation. Since all samples were incubated under the same conditions, it would be expected that dissolved oxygen levels in the waters would be about the same. In addition, no correlation was found between dissolved oxygen content and hydrazine degradation in the waters (Reference 4). Our results indicate that microbial activity and buffer capacity of the waters play a key role in the degradation of hydrazine sulfate. In light of the rapid hydrazine degradation and high bacterial populations of the Santa Fe River water, a large number of specific degraders may be present in this water. In addition, the high buffer capacity of this water may reduce hydrazine toxicity toward bacteria.

The Achromobacter sp. in the waters of Santa Fe River and Lake Alice, as well as in tap water and distilled water, also had a high capacity to degrade nonsalt forms of hydrazine (the monohydrate in this case) (Table 23). In fact, the bacterium in these waters degraded the nonsalt form better than the salt form. The average pH values of the four waters supplied with hydrazine monohydrate at 17, 34, and 68 µg/mL were 8.42 ± 0.24, 8.58 ± 0.20, and 8.73 ± 0.17, respectively. At concentrations of 17 and 34 µg/mL, hydrazine monohydrate exerted no adverse effect on degradative activity of the Achromobacter sp. At these concentrations 92 to 95 percent of the nonsalt form was degraded. At a concentration of 68 µg/mL, somewhat less degradation activity was observed for all four waters. No degradation was observed, however, in the waters without adding the bacterium during 2 hours of incubation. Also, no degradation was observed after adding autoclaved cells of Achromobacter sp. to the waters.

The Achromobacter sp. enhanced hydrazine degradation in all water samples but did not do so in Arredondo soil during one hour of incubation. *Degradation of hydrazine in soil* treated and untreated with Achromobacter sp. was not significantly different (±1 percent level) at hydrazine concentration of 100 µg/g to 500 µg/g. However, addition of the bacterium to soil suspensions (10 percent soil) did enhance hydrazine degradation (Table 24). The degree of degradation enhancement by the bacterium in the soil suspensions was not as great, however, as for the water samples.

Inability of the Achromobacter sp. to enhance hydrazine degradation in soil may be in part due to hydrazine toxicity to the bacterium and in part due to a soil surface effect. The soil-water content used in this study was 0.1 mL/g of soil. Since hydrazine is highly water-soluble, at a concentration of 100 µg/g of soil the concentration of hydrazine in the aqueous phase would be 1000 µg/mL (assuming no adsorption to soil surfaces). At this concentration the activity of Achromobacter sp. would be inhibited (Reference 39). Soil surfaces also may hinder the activity of the organism. This could result in loss of its enzyme activity, or in physical separation of the organism from the target chemical. In short, Achromobacter sp. may have potential as an agent for the detoxification of hydrazine in contaminated waters and liquid wastes.

TABLE 23. DEGRADATION OF HYDRAZINE MONOHYDRATE IN WATERS SUSPENDED WITH CELLS OF ACHROMOBACTER SP.[a]

Water	Initial N_2H_4 (µg/mL)	Percent Reduction after 2 hours	Initial N_2H_4 (µg/mL)	Percent Reduction after 2 hours
Santa Fe River	17	94	34	94
Lake Alice	17	94	34	94
Tap water	17	95	34	93
Distilled water	17	94	34	92
Santa Fe River	68	51		
Lake Alice	68	54		
Tap water	68	54		
Distilled water	68	49		

[a]Bacterial cells were added at 2.2 ± 0.2 mg/mL.

TABLE 24. HYDRAZINE DEGRADATION IN SOIL SUSPENSIONS WITH AND WITHOUT ADDED ACHROMOBACTER SP.[a]

Soil Suspension	Initial N_2H_4 ($\mu g/mL$)	Percent Reduction after 2 hours	Initial N_2H_4 ($\mu g/mL$)	Percent Reduction after 2 hours
		Unbuffered		
Unsterile + cells[b]	25	56	50	45
Sterile + cells[b]	25	57	50	45
Unsterile	25	45	50	30
Sterile	25	41	50	30
		Buffered		
Unsterile + cells[b]	25	92	50	81
Sterile + cells[b]	25	91	50	83
Unsterile	25	85	50	59
Sterile	25	85	50	64

[a] 0.5 g of Arredondo soil in 5 mL of unbuffered or buffered distilled water.

[b] Weight of cells 2.2 ± 0.2 mg/mL.

TABLE 25. EFFECT OF HYDRAZINE ON SOIL RESPIRATION IN ARREDONDO SOIL TREATED WITH HYDRAZINE AT 0, 2.5, 25 AND 125 µg/g.

Days	Rate of CO_2 Production (mg CO_2-C/100g soil/day)			
	Hydrazine (µg/g)			
	0	2.5	25	125
1	2.79±0.11	2.41±0.07	1.99±0.10	1.64±0.26
3	1.55±0.02	1.66±0.05	1.76±0.04	1.68±0.04
7	0.70±0.04	0.82±0.08	0.80±0.03	0.86±0.09
11	0.72±0.11	0.75±0.11	0.65±0.06	0.73±0.10
14	0.54±0.04	0.50±0.02	0.49±0.05	0.69±0.09
18	0.51±0.10	0.41±0.02	0.41±0.06	0.61±0.10
21	0.50±0.10	0.41±0.01	0.44±0.06	0.52±0.03
Total[a]	16.73	16.38	15.74	17.43
		$LSD^{b}_{0.05}$ = 1.93		

[a]Total cumulative CO_2-C(mg) production in 21 days.

[b]$LSD_{0.05}$ least-significant difference at the 5% level.

d. Effect on Soil Microbial Activity

Soil respiration (total CO_2 evolution) in hydrazine-treated soils was initially inhibited, with the degree of initial inhibition progressively increasing as hydrazine concentration increased (Tables 25 and 26). However, the inhibition was temporary. In fact, not only had all samples recovered from the inhibition within 2 days, but CO_2 production was actually enhanced. CO_2 production then levelled off after 6 or 7 days. Total cumulative CO_2 production in all treatments was not significantly different ($p = 0.05$) after 21 days.

TABLE 26. EFFECT OF HYDRAZINE ON SOIL RESPIRATION IN ARREDONDO SOIL TREATED WITH HYDRAZINE AT 0, 250 AND 500 µg/g.

Days	Rate of CO_2 Production (mg CO_2-C/100g soil/day)		
	Hydrazine (µg/g)		
	0	250	500
1	2.52±0.04	1.43±0.15	1.27±0.08
2	1.77±0.03	1.97±0.13	1.57±0.06
3	1.31±0.02	1.47±0.13	2.14±0.06
6	0.85±0.06	0.89±0.05	1.09±0.02
10	0.58±0.05	0.62±0.06	0.61±0.02
14	0.50±0.01	0.56±0.04	0.52±0.01
17	0.53±0.05	0.57±0.04	0.55±0.06
21	0.43±0.02	0.52±0.10	0.47±0.06
Total[a]	15.78	16.05	16.30
	$LSD^{b}_{0.05}$ = 1.64		

[a]Total cumulative CO_2-C(mg) production in 21 days.

[b]$LSD_{0.05}$ least-significant difference at the 0.05 percent level.

Similar to CO_2 production, bacterial populations in hydrazine-treated soils were also reduced initially (Table 27), although fungal populations were not affected. The reduction of bacterial populations appeared to be the principal cause of the inhibition in CO_2 evolution. For the 100 µg/g treatment, bacterial populations quickly recovered. This reflected the fact that, at this concentration, hydrazine was completely degraded within 1 day (Table 16). In fact, bacterial populations were enhanced in 7 days and remained larger than the control treatment thereafter. In contrast, bacterial populations for the 500 µg/g treatment were at least 10 times smaller than for the control treatment throughout the 28 days of incubation. After 7 days fungal populations for the 100 and 500 µg/g treatments were significantly larger than

TABLE 27. EFFECT OF HYDRAZINE ON BACTERIAL AND FUNGAL POPULATIONS IN ARREDONDO SOIL.

Concentration of hydrazine (µg/g)	Days 1	7	14	21	28
	Bacteria (cfu/g[a] x 10^{-6})				
0	13.92	15.45	9.83	7.63	7.21
100	1.35	24.90	42.60	25.80	16.40
500	0.82	0.68	0.91	0.44	0.56
	Fungi (cfu/g[a] x 10^{-4})				
0	1.05	2.26	2.70	2.22	2.61
100	0.87	6.71	9.36	9.59	9.70
500	1.00	3.85	11.44	9.49	7.84

[a]cfu/g colony forming units per gram of soil.

for the control treatment. Because of the magnitude of the reduction in bacterial populations for the 500 µg/g treatment, not only nitrifying bacteria, denitrifying bacteria, anaerobic bacteria (Reference 27), and Enterobacter cloacae (Reference 29) would be killed, but many other bacteria could be killed as well.

Hydrazine at concentrations of 10 and 100 µg/g did not exert any adverse effect on nitrification after 49 days for the Arredondo soil (Table 28). However, nitrification did not take place to a significant extent in the 500 µg/g treatment. As mentioned above, bacterial populations in this treatment were profoundly reduced, and nitrifying bacteria most likely would be killed at this concentration. Our results suggest that, at concentrations of 100 µg/g and lower, hydrazine exerts no adverse effect or only a short, temporary effect on soil microbial activity.

TABLE 28. EFFECT OF HYDRAZINE ON NITRIFICATION IN ARREDONDO SOIL.

Hydrazine concentration (µg/g)	NH_4^+-N[a] (µg/g)	NO_3^--N[a] (µg/g)
0	79.6±0.9	60.6±0.8
10	69.3±11.1	60.9±2.0
100	82.7±0.5	53.8±0.5
500	123.1±3.0	12.6±0.2

[a]Results at 49 days

2. Monomethylhydrazine

a. Effect on Soil Microbial Activity

MMH at concentrations ranging from 10 to 500 µg/g did not inhibit soil respiration in Arredondo soil. Unlike hydrazine, which initially inhibited total CO_2 production by soil, total CO_2 evolution was actually enhanced initially by the treatment with MMH (Tables 29 and 30). In fact, initial total CO_2 production became progressively larger as MMH concentration was increased. Total cumulative CO_2 production during 21 days for all of the MMH-treated samples was significantly higher than for the untreated samples.

Total aerobic bacterial populations and total fungal populations were also not inhibited by 10 µg/g of MMH (Table 31) and, at 100 µg/g and larger, total aerobic bacterial populations (Tables 31 and 32) were actually significantly larger than for the control treatments throughout the entire 21 days of incubation, with total bacterial populations becoming progressively larger as MMH concentration was increased. At 100 µg/g, total fungal populations were either not affected or were increased as well, though total fungal populations in soil treated with 200 and 500 µg/g of MMH were significantly but not severely reduced. The effect of 500 µg/g of MMH was in contrast to that of hydrazine, which severely reduced bacterial populations but enhanced fungal populations.

TABLE 29. TOTAL CO_2 PRODUCTION FROM ARREDONDO SOIL TREATED WITH MMH AT 0, 10, 50 AND 100 µg/g.

Days	Rates of CO_2 Production (mg CO_2 - C/100g soil/day)			
	MMH (µg/g)			
	0	10	50	100
0-1	4.61 ± 0.12	5.00 ± 0.15	5.24 ± 0.18	5.56 ± 0.15
1-2	3.66 ± 0.14	3.82 ± 0.14	4.09 ± 0.18	4.17 ± 0.13
2-5	2.00 ± 0.06	2.13 ± 0.03	2.41 ± 0.34	2.40 ± 0.29
5-7	1.47 ± 0.09	1.57 ± 0.06	1.64 ± 0.11	1.65 ± 0.09
7-11	1.16 ± 0.05	1.25 ± 0.00	1.33 ± 0.09	1.34 ± 0.05
11-14	1.05 ± 0.01	1.21 ± 0.15	1.14 ± 0.06	1.20 ± 0.04
14-18	0.88 ± 0.04	1.09 ± 0.13	1.01 ± 0.05	1.05 ± 0.05
18-21	0.78 ± 0.04	0.92 ± 0.09	0.90 ± 0.03	0.93 ± 0.04
Total (mg)	32.80	36.29	37.19	38.16
	$LSD_{0.05}$ = 2.32			

b. Degradation in Soil

MMH disappeared rapidly from both nonsterile and sterile soils. MMH at 10 µg/g completely disappeared from nonsterile and sterile soils in 30 minutes. Even at 100 and 500 µg/g, only 41.8 and 67.4 percent of the applied MMH were detected in nonsterile soils (Table 33), respectively, 30 minutes after application. After 48 hours, only small amounts of MMH remained in either nonsterile or sterile samples. The percentage of MMH remaining in sterile soils was consistently slightly higher than in nonsterile soils. This suggested that chemical degradation was the most important factor contributing to the disappearance of MMH from soil. Biological degradation also contributed to the disappearance of MMH, though much less significantly.

Despite the fact that biological degradation played only a minor role in the disappearance of MMH from soil, it was found that substantial amounts of ^{14}C-MMH in nonsterile soil were mineralized to CO_2. The evolved and trapped ^{14}C-activity in the KOH traps was principally $^{14}CO_2$, since little ^{14}C-activity remained in the supernatants after precipitation with $BaCl_2$ (Table 34). Degradation of MMH to CO_2 is a microbial process. After 9 days of incubation, 46.5 and 42.6 percent of the applied ^{14}C-activity were trapped in KOH for Arredondo soil treated with 100- and 500-µg/g of ^{14}C-MMH, respectively. More than 95 percent of the trapped ^{14}C-activity was found to be associated with $^{14}CO_2$. Furthermore,

TABLE 33. MMH IN NONSTERILE AND STERILE ARREDONDO SOIL.

Hours	MMH (percent remaining)			
	100 µg/g		500 µg/g	
	Nonsterile	Sterile	Nonsterile	Sterile
0.5	41.8	49.8	67.4	70.6
4	8.7	ND[a]	23.4	ND
24	1.6	5.3	1.3	4.3
48	0.7	2.3	0.6	1.2

[a]Not determined.

TABLE 34. ^{14}C-ACTIVITY EVOLVED AND TRAPPED IN KOH FROM ARREDONDO SOIL TREATED WITH ^{14}C-MMH.

Days	Percent of applied ^{14}C-activity	
	100 µg/g	500 µg/g
1	12.3[a] (0.6)[b]	6.8[a] (0.4)[b]
2	32.3 (1.7)	14.5 (1.2)
3	38.7 (1.9)	27.1 (1.4)
6	44.1 (2.1)	39.7 (1.6)
9	46.5 (2.2)	42.6 (1.9)

[a] ^{14}C-activity in KOH before addition of $BaCl_2$.

[b] ^{14}C-activity in KOH after addition of $BaCl_2$.

6.9 and 4.7 percent of the applied ^{14}C-activity in the 100- and 500-µg/g treatments could be extracted with 0.1N HCl, respectively, and 26.9 and 28.8 percent of applied ^{14}C-activity remained in the extracted 100 and 500 µg/g treated soils, respectively. Total ^{14}C-recoveries for the 100 and 500 µg/g treatments were 80.3 and 76.1 percent, respectively. MMH at 25°C has a vapor pressure of 49 mm Hg (Reference 40), which is somewhat higher than the vapor pressure of water. Hence, at least a part of the unaccounted for ^{14}C-activity had been lost from volatilization. No $^{14}CO_2$ was evolved from sterile soil treated with ^{14}C-MMH.

Although MMH disappeared rapidly from both nonsterile and sterile soils, our findings suggest that the nature of MMH degradation in nonsterile soil may be different from that in sterile soil. Alternatively, it is possible that MMH in both nonsterile and sterile soils was initially oxidized chemically to an oxidation product, while the product was subsequently degraded microbiologically to CO_2 in nonsterile soil. The product in sterile soil was not degraded further.

The enhancement of CO_2 production in MMH-treated soils was in part due to the increase in aerobic bacterial populations and in part due to the mineralization of MMH to CO_2. Since degradation of MMH to CO_2 is principally microbial, it is possible that MMH-degrading microorganisms can be isolated from soil and used for detoxification of contaminated soils, water, and wastes.

c. Microbial Degradation

MMH at 25 and 100 µg/mL was rapidly degraded by the growing cultures of both Achromobacter sp. and Pseudomonas sp. (Figures 48 and 49). Both bacteria required a second carbon source (glucose) and a second nitrogen source (ammonium nitrate) for growth. Since only small amounts of MMH were degraded in the autoclaved cultures, degradation of MMH in the growing cultures was principally microbial. MMH at both concentrations was rapidly degraded by the growing culture of Pseudomonas sp. without a lag period. This was possibly due to either a high initial inoculum level or a high initial degradative enzyme activity. Higher MMH concentrations appeared to prolong the lag phase of bacterial growth. Although the Achromobacter sp. degraded MMH, growth of this organism in the presence of MMH at the end of the incubation period (96 hours) had not reached the same level as in the absence of MMH.

Neither Achromobacter sp. nor Pseudomonas sp. degraded MMH to CO_2. Although MMH was completely degraded in 4 days by both bacteria, the majority of the applied ^{14}C-activity still remained in the culture fluids (Table 35). Less than 4 percent of the applied ^{14}C-activity was found in the KOH traps. No change of radioactivity in the KOH traps was observed before and after acidification of the KOH. This indicated that the trapped ^{14}C-activity was not associated with $^{14}CO_2$. The trapped ^{14}C-activity probably consisted of ^{14}C-MMH and volatile metabolites. MMH is somewhat more volatile than water (Reference 40). Total ^{14}C-recoveries for all treatments were near 100%.

TLC-autoradiographic assays confirmed the results of the colorimetric determinations, namely that MMH had completely disappeared from 4-day-old cultures of the two bacteria. The TLC-autoradiographic assays also revealed that MMH was degraded to water-soluble polar metabolites (R_f value = 0).

Resting cell suspensions of the two organisms had a high capacity to degrade MMH (Table 36). During 1 hour of incubation, MMH at concentrations under 50 µg/g was either completely or near-completely degraded. Even at 117 µg/g, 57 and 34 percent of the MMH were degraded in cell suspensions of the Achromobacter sp. and the Pseudomonas sp.,

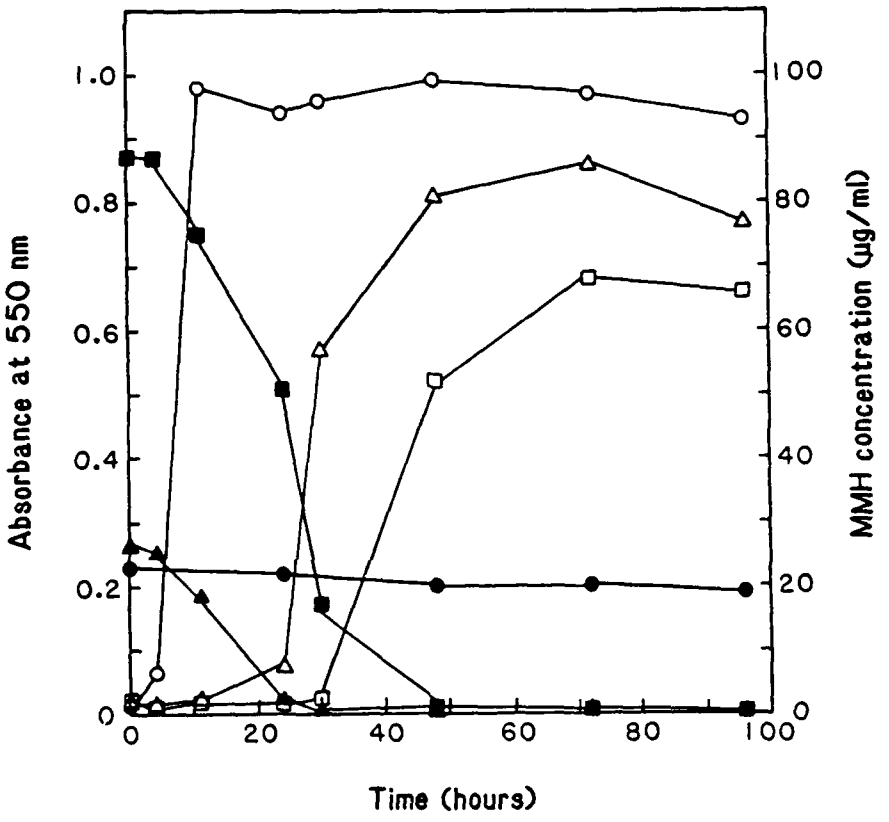


Figure 48. MMH Degradation and Growth of Achromobacter sp. Designations: ○, △, and □, Absorbance of Culture Fluids with Initial MMH Concentrations of 0, 27 and 88 µg/mL, Respectively; ▲ and ■, MMH Concentrations in Culture Fluids; and ●, MMH Concentration in the Culture-free Medium.

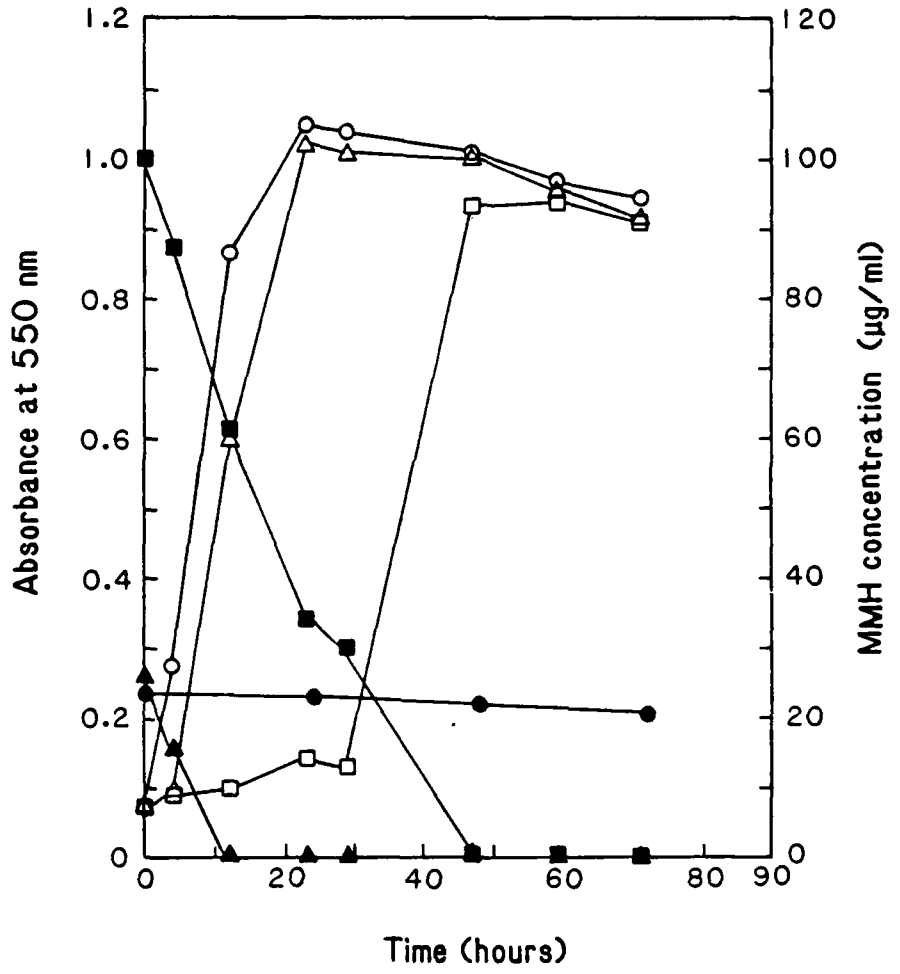


Figure 49. MMH Degradation and Growth of Pseudomonas sp. Designations: ○, △, and □, Absorbance of Culture Fluids with Initial MMH Concentrations of 0, 27 and 100 µg/mL, Respectively; ▲ and ■, MMH Concentrations in Culture Fluids; and ●, MMH Concentration in the Culture-free Medium.

TABLE 35. ^{14}C-ACTIVITY IN CULTURE FLUIDS OF ACHROMOBACTER SP. AND PSEUDOMONAS SP., AND IN KOH TRAPS AFTER 4 DAYS OF INCUBATION.

MMH (µg/g)	Percent of applied ^{14}C-activity		
	^{14}C-Activity Remaining in Culture Fluids	^{14}C-Activity Trapped in KOH	Recovery
		Achromobacter sp.	
25	94.4	4.8	99.2
100	94.0	4.7	98.7
		Pseudomonas sp.	
25	96.4	4.4	100.8
100	96.8	4.4	101.2

TABLE 36. DEGRADATION OF MMH IN LIVE AND AUTOCLAVED CELL SUSPENSIONS OF ACHROMOBACTER SP. AND PSEUDOMONAS SP.[a]

Cell Suspension	Initial MMH (µg/g)	Percent Reduction after 1 hour
	Achromobactar sp.	
Live[b]	22	100
Live[b]	41	100
Live[b]	81	83
Live[b]	117	57
Autoclaved[c]	21	3
Autoclaved[c]	64	3
	Pseudomonas sp.	
Live[d]	21	100
Live[d]	45	96
Live[d]	86	63
Live[d]	117	34
Autoclaved[e]	22	4
Autoclaved[e]	67	6

[a]Cell suspensions were incubated at 25°C for 1 hour.

[b]Cell dry weight 4.0 mg/mL.

[c]Cell dry weight 4.0 mg/mL.

[d]Cell dry weight 4.1 mg/mL.

[e]Cell dry weight 4.3 mg/mL.

respectively. Less than 6 percent of the MMH disappeared from autoclaved cell suspensions. This suggests that degradation in the resting-cell suspensions was principally microbial. Since a substantial amount of ^{14}C-MMH in soil is mineralized to $^{14}CO_2$, and since such degradation is microbial, microorganisms that utilize MMH as a sole source of carbon for growth may exist. Efforts are being made to isolate such microorganisms from soil.

D. CONCLUSIONS

Both hydrazine and MMH are rapidly degraded in nonsterile and sterile soils and autooxidation is the major factor in contributing to the disappearance of the chemicals. Biological degradation also contributes to the disappearance, but is a minor factor.

Microorganisms that have capacity to degrade hydrazine and MMH exist in soil. Three soil bacteria, an Achromobacter sp., a Pseudomonas sp. and a Bacillus sp. are found to have capacity to degrade hydrazine. The Achromobacter sp. and the Pseudomonas sp. also cometabolically degrade MMH.

Despite the fact that Achromobacter sp. and Pseudomonas sp. can not utilize hydrazine as a sole source of nitrogen as well as can not utilize MMH as a sole source of carbon for growth, these two bacteria also do not degrade MMH to its final oxidation products CO_2 and H_2O. The organisms may have potential for use in the detoxification of hydrazine - and MMH-contaminated soils, water, and wastes. In this case, the nature of the water soluble polar metabolites of MMH need to be determined.

Due to rapid degradation of hydrazine and MMH in soil, bacterial and fungal populations in soil are less sensitive to the chemicals than to the soil axenic bacterial cultures. In fact, hydrazine enhances fungal populations in soil, and MMH enhances soil respiration and bacterial populations in soil.

E. RECOMMENDATIONS

Similar research for determination of degradation rates of hydrazine and MMH in soil and their toxicity to soil microbial activity should also be conducted for 1,1-dimethylhydrazine (UDMH).

Microorganisms that have capacity to degrade UDMH to nontoxic products should be isolated from soil and water. Attempts should be made to use the organisms to detoxify UDMH from contaminated soils, waters and wastes.

A microbial system that can completely defoxify hydrazine, MMH and UDMH in contaminated soils, waters and wastes should be developed.

A research is needed to study the fate of hydrazine fuels in subsurface soils and groundwater, and microbial degradation of the chemicals in subsurface soils and groundwater.

A microbial system that can detoxify hydrazine fuels in vadose zone soils and groundwater should be developed.

SECTION V

SOIL PHYSICS AND MODEL DEVELOPMENT

A. INTRODUCTION

Hydrazine is a highly reactive diamine NH_2-NH_2 (or N_2H_4) which is a strong reducing agent in alkaline aqueous solution (reference 40). In alkaline and neutral solutions hydrazine undergoes rapid chemical degradation by the process of autooxidation. In acid solution hydrazine is hydrolyzed to give the monovalent cation $N_2H_5^+$ which is not susceptible to autooxidation. Schmidt (reference 40) states that acidified hydrazinium salt solutions can be stored for months without change in composition. Thus, the potential for contamination of groundwater in the event of accidental spillage of hydrazine onto a soil or subsurface leakage of hydrazine from underground storage tanks depends greatly upon soil pH and the capacity of the soil to buffer against rapid changes in pH. Soils that occur in a humid climate such as that in Florida are typically acidic with pH commonly ranging from 4 to 6. Acid sandy soils in Florida typically have low acid buffering capacities due to low contents of alumino-silicate clay minerals, iron and aluminum oxides, organic matter, and other chemically reactive components.

This portion of the study was designed to determine if $N_2H_5^+$ would move through columns of water-saturated soil during steady flow and to develop a transport model to describe the movement and fate of $N_2H_5^+$ in soil columns. Aqueous solutions of $N_2H_5^+$ were miscibly displaced through a series of soil columns to generate experimental information to evaluate the mobility of $N_2H_5^+$ during water transport in soil. That data was also used to evaluate proposed models.

Previous research results (references 1, 44, 47) have shown N_2H_4 and $N_2H_5^+$ molecules to be highly reactive in soil-water systems. Under acidic conditions the hydrazinium form tends to dominate in the solution phase of the soil. In acidic water $N_2H_5^+$ is chemically stable and therefore persistent; whereas in acidic soil-water systems $N_2H_5^+$ ions react physically, chemically, and biologically with reactive soil constituents such as organic matter, alumino-silicate clay minerals, and sesquioxides. Major reactions known to influence hydrazinium in acid soil-water systems include the following:

	Reaction	Effect Upon Potential Contamination of Groundwater
1.	Ion exchange or physical sorption between $N_2H_5^+$ ions and exchangeable soil cations	1. Reversible removal of $N_2H_5^+$ ions from the mobile soil solution
2.	Complexation or nonspecific sorption of $N_2H_5^+$ ions with cations adsorbed onto reactive soil constituents	2. Partially-reversible removal of $N_2H_5^+$ ions from the mobile soil solution
3.	Condensation reaction of $N_2H_5^+$ ions with carbonyl groups of humic components	3. Irreversible removal or chemisorption of $N_2H_5^+$ ions from the mobile soil solution
4.	Microbiological degradation of $N_2H_5^+$ by soil microorganisms	4. Destructive irreversible removal of $N_2H_5^+$ ions from the soil-water system

Hydrazinium is not susceptable to chemical degradation by autooxidation under acid conditions.

The complexity of $N_2H_5^+$ reactions with soil organic matter is clearly illustrated by results published by Isaacson and Hayes (Reference 47). They used a continuous-flow method to obtain $N_2H_5^+$ sorption isotherms for hydrogen-, aluminum-, and calcium-exchanged humic acid preparations (i.e., extracts from histosols) in pH 4 aqueous suspensions. Mean sorption residence times ranged from 7 to 270 minutes for the liquid flow rates imposed on the reaction cells. Ion exchange, chemisorption and nonspecific sorption were concluded to be the major sorption processes in those systems. For a specific case where 2.30 mol $N_2H_5^+$ was sorbed per

kg of Ca^{2+}- humate, relative amounts extracted by water and 0.1 M NaCl solution were 17 and 43 percent, respectively. Approximately 26 percent of the sorbed $N_2H_5^+$ was irreversibly sorbed and 14 percent was not accounted for in the humic residue. Thus, only 43 percent of the sorbed $N_2H_5^+$ was desorbed by the physical process of ion exchange. Other processes contributed to 26 percent of the sorbed $N_2H_5^+$ being tightly bound to the humic materials. The exchangeable cation species on the humic substances was also observed to influence overall sorption of $N_2H_5^+$. For example, $N_2H_5^+$ ions were more extensively held by H^+- humic acid than by Ca^{2+}- humate or Al^{3+}- humate. This observation was attributed to greater ability of $N_2H_5^+$ ions to exchange with H^+ than with Ca^{2+} and Al^{3+} ions on carboxyl exchange sites and to disrupt hydrogen bonds rather than divalent- and polyvalent-cation bridges between polymeric strands of the humic substances. As hydrazinium ions were sorbed by Ca^{2+}-humate, Ca^{2+} ions were desorbed; whereas sorption of hydrazinium ions by Al^{3+}-humate did not result in detectable desorption of Al^{3+} ions. The authors showed that ion exchange was involved in sorption of $N_2H_5^+$ ions by Al^{3+}-humate even though Al^{3+} ions were not detected in the equilibrium solution. Changes in differential enthalpy of the sorption of $N_2H_5^+$ were such that the affinity by which hydrazinium ions were held by Ca^{2+}-humate increased as the quantity of sorbed $N_2H_5^+$ increased. In contrast, the affinity by which $N_2H_5^+$ ions were held by H^{+1}-humic acid decreased with increasing quantity of sorbed $N_2H_5^+$. Thus, the influence of chemisorption (i.e., condensation reaction) and non-specific sorption appeared to be lowest for small concentrations of $N_2H_5^+$ in solution for the Ca^{2+}-humate and appeared to increase as the concentration increased. For sorption of $N_2H_5^+$ by H^+-humic acid, the influence of chemisorption and nonspecific sorption appeared to be maximal at low $N_2H_5^+$ concentrations and to decrease as the concentration increased. These findings are particularly pertinent to sorption of $N_2H_5^+$ ions in acid sandy soils where organic matter exists as coatings on clay minerals as well as discrete particles. Although the cation exchange capacity of such soils are typically small, the fractional contribution of organic matter to the exchange complex is often large.

Microbial degradation of N_2H_4 and $N_2H_5^+$ in soil and in water has been

investigated by Ou (Reference 39), Ou and Street (Reference 55) and Ou and Street (Reference 3). Under aerobic, water-unsaturated soil conditions degradation proceeded more rapidly in Arredondo sand than in water. Degradation rates in the Arredondo soil were observed to decrease as the quantity of hydrazine applied increased. They reported that hydrazine applied to Arredondo soil at concentrations of 10, 100, and 500 $\mu g\ g^{-1}$ completely disappeared within 2, 24, and 168 hours, respectively. Only 20 percent of the hydrazine disappearance was directly attributable to biological degradation. Presumably, the remaining 80 percent of the disappearance was due to combined effects of irreversible chemisorption and nonspecific sorption to soil components. For hydrazine concentrations of 100 $\mu g\ g^{-1}$ and lower, hydrazine exerted no adverse effect or only a short, temporary effect upon activity of soil nitrifying bacteria. A toxic effect was observed for concentrations of 500 $\mu g\ g^{-1}$. Thus, these results indicate that microbial degradation of hydrazine in aerobic Arredondo soil is greatest for low application rates of the chemical and decreases with increasing application rate.

The net effect of ion exchange, complexation, condensation, and microbial degradation processes upon the transport of $N_2H_5^+$ with water through the porous matrix of soils should be to minimize local concentration levels of $N_2H_5^+$ in the mobile solution phase so as to retard the overall migration rate of the chemical. A method for partial removal of irreversibly retained hydrazine by acid extraction of soil was reported (Reference 49) recently. In the event of accidental spillage of hydrazine onto the soil surface or unsuspected leakage of effluent from underground storage tanks, the soil profile provides a hydrological connection to underlying groundwater. Thus, as aqueous hydrazine solutions move into and through the soil, the four major reactions that influence hydrazine concentrations in the solution phase provide protection against contamination of groundwater. A search of the published literature revealed a severe lack of experimental results for hydrazine movement in soil from either laboratory soil columns (Reference 44) or from intact soil profiles at field sites. This lack of experimental information concerning hydrazine mobility in soils is somewhat surprising since hydrazine hydrate was discovered 100 years ago (Reference 58) and

is currently used for multiple purposes. Clearly, such information is needed to evaluate the environmental impact of accidental spills or storage tank leakage.

B. LABORATORY INVESTIGATION OF HYDRAZINIUM REACTIONS AND TRANSPORT IN COLUMNS AND STIRRED SUSPENSIONS OF SOIL

1. Objectives

Soil columns and stirred aqueous slurries of soil were used to investigate selected major factors which influence the persistence and mobility of $N_2H_5^+$ applied to a coarse-textured soil. Factors investigated include concentration of $N_2H_5^+$ in applied aqueous solution, method of solution application, and pore water velocity in soil columns during transport.

2. Experimental Methods and Materials

Data needed to select reactions and transport processes, to rank them in order of importance, and to evaluate them were determined from chemical analysis of aliquots of effluent from saturated soil columns and from stirred aqueous suspensions of three sequential profile horizons of Arredondo fine sand.

a. Soil Properties

Samples from Ap, E1, and E2 horizons of Arredondo fine sand were obtained from a site (0.4 mile east of state road 241 and 0.6 mile north of state road 222) in NW Alachua County, Florida. Arredondo fine sand is a loamy, siliceous, hyperthermic, Grossarenic paleudult (reference 60), and is typical of the well-drained soils of Florida. At the collection site the Ap horizon extended from the surface to a depth of 20 cm. The E1 horizon was found between 20- to 80 cm depths and the E2 horizon occurred between 80- to 120 cm depths. These horizons were visually distinguished from one another in the soil profile. Properties

of the horizons which affect their capabilities for reacting with and transporting hydrazinium during water flow include texture, organic carbon content, acid-buffering capacity, pH, chemical composition, and cation exchange capacity.

The distribution of sizes for mineral particles in a soil matrix has a significant effect on the moisture and chemical retention properties of the soil. Soils high in percentage sand-size particles tend to retain water poorly and are relatively nonreactive chemically when compared to soils higher in contents of smaller silt- and clay-size particles.

Particle size analysis (mechanical analysis) for mineral soil components was performed on samples of the three horizons by the pipette method of Gee and Bauder (Reference 46). Samples were suspended in distilled water and dispersed with sodium hexametaphosphate. The supernatant was decanted and allowed to settle in a constant temperature water bath from which aliquots were removed by pipette at a depth and time corresponding to settling velocity determined by Stoke's Law. Samples were dried and weighed to determine percentage clay. Remaining material was washed, dried, and filtered through 16-, 32-, 60-, 150-, and 325-mesh U.S.A. Standard Testing sieves to determine various sand fractions. Percentage silt was determined by subtracting sand and clay percentages from 100.

Organic carbon contained in the organic fraction of soil consists of cells of microorganisms, plant and animal residues in various stages of decomposition, stable humus synthesized from residues, and highly carbonized compounds such as charcoal, graphite, and coal (Reference 54). The determination of the amount of organic material present in the soil is important since many groundwater contaminants including hydrazine (Reference 47) react with organic material.

Percent organic carbon was determined by dry combustion in an induction furnace (LECO Model No. 523-300). A sample with known weight was placed in a ceramic crucible with iron and copper metal accelerator added. The sample was heated inside an enclosed combustion tube through which oxygen was passed. All of the carbon in the sample was oxidized to CO_2. Small particles were removed in a dust trap, and

sulfur was absorbed in a sulfur trap, leaving only CO_2 and oxygen. The CO_2-oxygen volume was measured in a buret held at constant temperature and corrected for pressure. The mixture was passed through a solution of KOH in another vessel which absorbed all the CO_2. The oxygen was brought back to the original buret, and the volume of CO_2 was determined by subtraction from the previous volume.

The dry combustion method described here determines total carbon present in the soil. Total carbon is the sum of both organic and inorganic carbon. Inorganic carbon is found in carbonate materials such as calcite, dolomite, and soluble carbonate salts, and is not generally found in well-leached soils of low pH (Reference 54). In such soils total carbon content is equivalent to organic carbon content.

The transport of water and soluble chemicals through water-saturated soil is dependent on physical, microbiological, and chemical processes. The chemical and microbiological processes act to retard or degrade the chemical in solution as it is moved down gradient under the influence of physical flow processes. Hydrodynamic dispersion is an important physical process which incorporates diffusion gradients and velocity distributions within soil pores. Dispersion coefficients corresponding to Darcy flow rates of 0.5 (1.4×10^{-6} m s^{-1}) and 5.0 (14.0×10^{-6} m s^{-1}) cm h^{-1} were determined by using the derivation of Kirkham and Powers (Reference 48) from data obtained by passing a pulse of tritiated water (10,000 cpm in 0.01N $CaCl_2$) through the soil columns. A scintillation fluid (Scintiverse II) was added to samples of column effluent and concentrations of ^3H_2O were determined using a Liquid scintillation counter. Kirkham and Powers (Reference 48) differentiate the complementary error function (erfc) mathematical solution to the conservative solute transport equation

$$C/C_o = 0.5 \text{ erfc } [\upsilon L(1-p)/4Dp] \qquad [1]$$

to obtain the slope m of the breakthrough curve (BTC) at p = 1 (or C/C_o = 0.5) where C is the solute concentration in the effluent, C_o is the solute concentration of the applied influent solution, p is the number of pore volumes of effluent, υ is the pore water velocity, D is the hydro-

dynamic dispersion coefficient, and L is the soil column length. The dispersion coefficient D can thus, be determined from the relationship

$$D = \upsilon L/4\pi m^2. \qquad [2]$$

The slopes of breakthrough curves for tritiated water tracers were determined and substituted into equation [2] to calculate the dispersion coefficients for each horizon at each flow rate.

Soil pH is very important to the transport of hydrazine in the soil environment since hydrazine molecules are protonated to yield hydrazinium ions in an acidic aqueous environment. In aqueous solution at pH 7.96 (i.e., the pK_a of hydrazine) hydrazine and hydrazinium occur in equal proportions. The protonated hydrazinium ions are able to undergo ion exchange reactions on soil particle surfaces, having a potentially significant impact on the transport process.

The pH of each soil horizon was determined by placing 25 g of soil in a 100 mL beaker, adding 50 mL of 0.01 N $CaCl_2$, stirring for two minutes with a magnetic stirring-bar, and reading the pH of the suspension.

The acid-buffering capacity of the three horizons of Arredondo fine sand was measured by preparing titration curves giving pH for each addition of $Ca(OH)_2$. Five grams of soil and 25 mL of $CaCl_2$ were added to a beaker and stirred for 3 minutes before the pH was read. $Ca(OH)_2$ was then added in equal increments and the pH noted. The $Ca(OH)_2$ was previously titrated with 0.01 N potassium phthalate to an end point of 0.0084 N.

Concentrations of calcium, aluminum, magnesium, iron, sodium, and potassium metals in the three soil horizons were determined by using flame atomic adsorption spectroscopy to chemically analyze acid-extractions for each soil material. Approximately 5-g samples of each soil horizon were first placed into a 50-mL polysulfone centrifuge tube into which was added 20 mL of 0.01 M HNO_3. The tubes were mechanically shaken for 4 hours at low speed, then centrifuged for 10 minutes at 10°C at 10,000 rpm with a 2,000 rpm per minute acceleration rate. Following centrifugation, supernatant in each tube was decanted into acid-washed

glass scintillation vials and analyzed using an atomic adsorption spectometer.

X-ray diffraction analysis was performed to determine the principle mineral species in each soil horizon. Approximately 500-g samples of each horizon were wet-sieved through a 0.0017-mm screen to remove sand particles. Approximately 100 mL of chlorox was added to the Ap horizon filtrate to oxidize soil organic material which tends to interfere with the X-ray diffraction process. After 2 days of oxidation time, 30 mL of 0.5 N HCl was added to flocculate the clay minerals. The suspension was allowed to stand for 1 day, then centrifuged at 16,000 rpm for 6 minutes. The centrifugation process was repeated 6 times, each time collecting the supernatant and resuspending the sediment in pH-10 distilled water. Approximately 100 mL of saturated NaCl was then added to flocculate clay material. An aliquot of clay suspension was placed on porous ceramic tiles, and 1 N $MgCl_2$, KCl, and glycerol were added to the tiles to allow differentiation of kaolinite from the smectite clays.

Published chemical analysis on samples of Arredondo fine sand by the Soil Conservation Service has revealed the soil to have a low Cation Exchange(Reference 60) Capacity (CEC). This analysis was confirmed by determining the CEC of the Ap, E1, and E2 horizons of Arredondo fine sand. Dilutions were made of a stock solution of 0.001 N $CaCl_2$. Forty mL of each dilution was placed in a polysulfone centrifuge tube along with 4-g of soil and shaken gently for 4 hours. The tubes were centrifuged for 10 minutes at 10,00 rpm, and the supernatant was analyzed for calcium. The decrease in calcium concentration in the supernatant was considered to be adsorbed onto the soil surface, and the plateau of the plot of calcium in solution versus adsorbed calcium was considered to be the exchange capacity. The CEC for the E2 soil was thus, determined to be approximately 4.0 mmol(+) kg^{-1} of soil.

b. Miscible Displacement Investigations with Soil Columns

Flow experiments were performed by initially pumpimg acidified 0.01 N $CaCl_2$ solutions for 24 hours into columns of air-dry soil in order to displace air from soil pores and saturate soil exchange sites with Ca^{+2} ions. Acidified 0.01 N $CaCl_2$ solutions containing a specified concentration of hydrazinium were pumped into each column as either a

pulse or as a continuous application in order to investigate the mobility of hydrazinium. Solutions were pumped with one of two different Darcy flow velocities (q) to simulate high and low velocities that might commonly be expected under field conditions. Effluent aliquots were collected using an automatic fraction collector and analyzed for hydrazinium, calcium, and pH. Results were graphed by plotting the relative concentration (C/C_o where C_o was the concentration in the applied solution) of the effluent component of interest against the number of pore volumes (p) of effluent. A schematic diagram of the equipment utilized in the miscible displacement experiments is shown in Figure 50.

Bulk samples of the three horizons of Arredondo fine sand were removed with a clean shovel from within each horizon sufficiently far removed from the horizon boundary to preclude contamination from above and below. Each bulk sample was sieved through a 2-mm mesh, on a flat tray, air-dried for 3 days, well-mixed, and then stored in plastic buckets for later use.

Cylindrical glass chromatography columns 30-cm long with 5-cm inside diameter (kontes No. 420800-3020) were hand packed to a bulk density of approximately 1.4 Mg m^{-3} for Ap soil and 1.6 Mg m^{-3} for E1 and E2 soils. These values for bulk density were determined earlier for Arredondo fine sand by the Soil Conservation Service.

Air initially present in packed columns of air-dry soil was displaced by introducing a flowing stream of helium gas into the top of each capped column for 1 hour prior to wetting up with acidified 0.01 $\underline{N}$ $CaCl_2$ solutions. Thus, O_2, N_2, and CO_2 gases originally present within soil pores were displaced out the bottom of the columns and replaced by nonreactive helium gas.

Helium gas in the pore space of the soil columns was then displaced with liquid by pumping helium-saturated, acidified 0.01 $\underline{N}$ $CaCl_2$ into the bottom of each column. Pumping was continued for at least 15 hours to assure a high degree of liquid saturation of the soil pore space and to insure replacement of exchangeable cations with Ca^{2+} on soil exchange sites. Aqueous 0.01 $\underline{N}$ $CaCl_2$ solution was deaerated for at least 2 hours by bubbling helium gas into continuously stirred flasks. Aqueous $CaCl_2$ solutions applied to columns of Ap, E1, and E2 soil were acidified

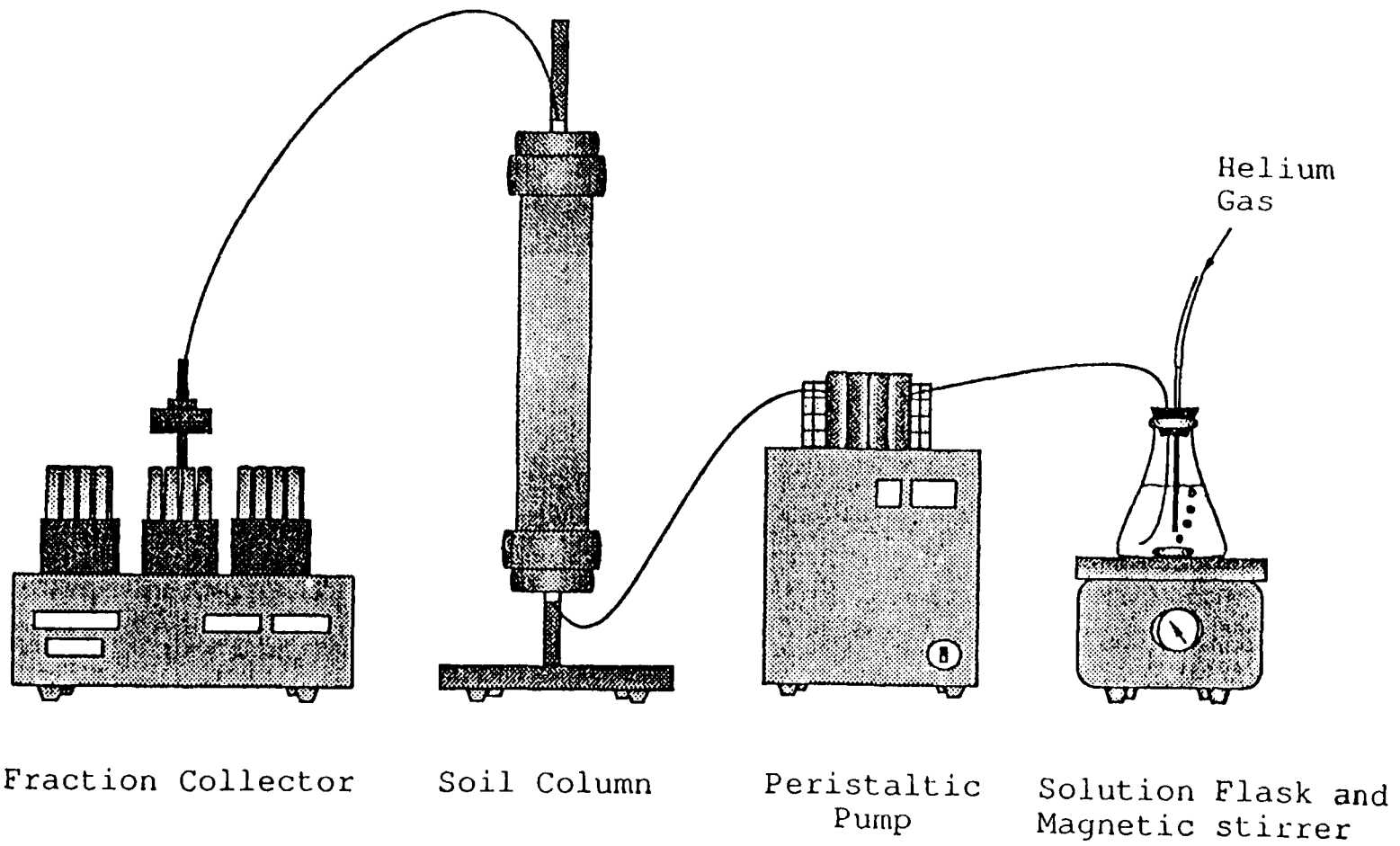


Fig 50. **Schematic Diagram of Soil Columns and Apparatus Used to Perform Miscible** Displacement **Experiments.**

by adding HCl to give pH values of 4.5, 5.0, and 5.1, respectively. Deaeration was insured by taking dissolved oxygen measurements with a calibrated dissolved oxygen probe. A 0.01 N $CaCl_2$ solution was used to approximate the ionic solution of natural groundwater in Florida which is dominated by the calcium cation. While 0.01 N is a stronger solution than is commonly found, it was used to insure saturation of soil cation exchange sites with calcium. The average soil water content for each column was determined by weighing the entire column before and after wetting up and assuming that the difference in the two weights was attributable to liquid.

Aqueous solutions were transferred from flasks to the soil columns by a Gilson peristaltic pump (Gilson Model Minipulse 2) through small diameter Tygon tubing. Pumping rates of 101.2 and 10.12 $cm^3\ h^{-1}$ were used to give Darcy flow velocities of 5 and 0.5 cm h^{-1} through the soil columns. Flow rate calibration for the pump used was made by adjusting pump speed to obtain the proper volume of column effluent per unit time as determined using a graduated cylinder.

Hydrazine hydrate ($N_2H_4 \cdot H_2O$) was used to prepare acidified 0.01 N $CaCl_2$ influent solutions with low (6.6 mg l^{-1}), medium (165 mg l^{-1}), and high (660 mg l^{-1}) concentrations C_o of hydrazinium ($N_2H_5^+$). Preliminary column work showed that pulses of 660 mg l^{-1} hydrazinium solutions applied to soil columns were sufficiently concentrated to overwhelm most of the soil attachment sites and sorption/degradation processes, while 165 mg l^{-1} hydrazinium solutions appeared to give definable BTC for hydrazinium. Application of 6.6 mg l^{-1} solutions of hydrazinium resulted in severe retardation and removal of $N_2H_5^+$ from the solution phase of the soil. The pH of the applied hydrazinium solutions was adjusted to pH 4.5 for Ap soil, 5.0 for E1 soil, and 5.1 for E2 soil by the addition of HCl, since each of the three horizons was acidic. The relative proportions of hydrazine and hydrazinium in aqueous solution can be determined from the relationship:

$$C/C_1 = 10^{7.96 - pH} \qquad [3]$$

where C and C_1 are the concentrations for N_2H_4 and $N_2H_5^+$, respectively,

and 7.96 is the pk_a value for the chemical reaction

$$N_2H_4 + H_2O \rightleftharpoons N_2H_5^+ + OH^- \qquad [4]$$

A plot of the relationship of the pH to the percent hydrazinium illustrates the mathematical relationship. In the soil pH range from 4.86 to 5.05 the hydrazine solution is found to be approximately 99.9 percent hydrazinium.

Acidified 0.01 N $CaCl_2$ solutions with a specified concentration C_o of hydrazinium at the two flow rates were pumped into the bottom of each column either as a pulse of 2-pore volume width or as a continuous input. Pulse inputs were followed by application of acidified 0.01 N $CaCl_2$ solution with $C_o = 0$. Application of influents as a pulse allowed an observation of ascending and descending limbs for each BTC for $N_2H_5^+$, Ca^{2+} and H^+ in column effluent, thus, giving valuable insight about the kinetic nature of reversible hydrazinium sorption processes. A breakthrough curve from a continuous input provided valuable information about irreversible processes of chemisorption and degradation.

A fraction collector (ISCO Model Retriever II) was positioned to collect effluent emerging from small diameter Tygon tubing connected to the top of the soil column. During the high flow rate, glass test tubes with 10-mm diameter were automatically moved under the emerging stream of effluent at 9-minute intervals, thus, collecting approximately 15 mL of effluent per tube, or about 13 tubes per pore volume of solute. One mL of 1 N HCl was added to alternate test tubes to ensure that hydrazine existed in the stable hydrazinium form. Nonacidified tubes were examined for pH and calcium concentrations.

For the low liquid flux ($q = 0.5$ cm h^{-1}) treatment, 12-mm diameter test tubes were used to collect fractions at 2-hour intervals. Each tube contained approximately 20 mL of effluent, or about 10 tubes per pore volume. Again, one mL of 1N HCl was added to alternate test tubes. Nonacidified tubes were examined for pH and calcium concentrations.

Hydrazine analysis was performed using a modification of the method of Waats and Chrisp (Reference 32). Small aliquots of col-

lected fractions were placed into a 25-mL volumetric flask along with 15 ml of p - Dimethylaminobenzaldehyde (PDBA). Hydrazine reacts with PDBA to form an intense orange color proportional to the concentration of hydrazine present. The solution was diluted and stabilized by the addition of 1N HCl to bring the volume up to 25 mL The color intensity was read with a spectrophotometer (Perkin Elmer, Coleman 54B) as percent transmission and then converted to absorbance. Hydrazine standard solutions were prepared with known concentrations and also read along with the effluent fractions. Effluent sample concentration was interpolated from the standard curve.

Calcium analysis of the effluent was performed by atomic adsorption spectrometry (Perkin Elmer flame spectrophotometer Model 460). Effluent samples were diluted 1 to 200 and absorbance was determined using a nitrous oxide flame. Interpolation of calcium concentration was made from a standard curve run at the same time as the samples of effluent. A determination of pH was made using a calomel electrode (Ross No. 2222 on an Orion meter No. 333) as effluent fractions were collected.

Analysis of the data obtained from each column effluent fraction was facilitated by plotting relative concentration (C/C_o) of $N_2H_5^+$ in the effluent versus the number of pore volumes of column effluent ($p = \beta/\beta_o$ where β is the cumulative volume of effluent and β_o is the volume of water present in the water-saturated soil column). At the time hydrazinium analysis was performed on samples of column effluent, analysis of concentrations (C_o) of influent solutions was performed for aliquots saved from flasks containing input solution. The hydrazinium concentration determined on each effluent fraction was divided by the concentration of the input solution to establish a relative scale with a maximum value of 1.

Breakthrough curves were also prepared for Ca^{2+} concentrations and pH of column effluent. The resulting breakthrough curves of solutes in column effluent reveal important information about the dynamics of physical and chemical interactions within the column as solute moves through the pores of the soil and emerges in the effluent.

3. Results and Discussion

Soil materials from the three upper horizons of Arredondo fine sand consisted of a dominant sand-size fraction and relatively small fractions of silt and clay (Table 37). From the standpoint of particle size distribution or texture, the E1 and E2 horizons have greater similarity to one another than they do to the Ap horizon. The 2.6 percent clay and 7.3 percent silt contents in the Ap horizon set it apart as significantly different than the two lower horizons.

TABLE 37. PARTICLE-SIZE DISTRIBUTION OF ARREDONDO FINE SAND

Soil Horizon	Sand Content						Silt Content	Clay Content
	Very Coarse (2-1mm)	Coarse (1-.5)	Medium (.5-.25)	Fine (.25-.1)	Very Fine (.1-.05)	Total (2-.05)	Total (.05-.002)	Total (<.002)
	(percent)							
Ap	0.0	3.0	21.6	57.3	8.2	90.1	7.3	2.6
E1	0.0	1.5	21.6	54.2	16.1	93.4	4.9	1.7
E2	0.0	3.3	29.6	50.1	11.5	94.5	3.7	1.8

The analysis of Arredondo fine sand revealed that the highest organic carbon content occurred in the surface Ap horizon and was much less in successively deeper soil horizons. The upper horizon contained 1.84 percent organic carbon compared to 0.34 and 0.14 percent for the E1 and E2 horizons, respectively. While these precentages are low, it is significant to note that the Ap horizon contains approximately six times as much organic carbon as the next lower horizon.

Hydrodynamic dispersion coefficients D obtained by displacing $^{3}H_2O$ solute tracer through columns of Ap, E1 and E2 Arredondo soil are given in Table 38 for Darcy liquid velocities of 0.5 and 5 cm h^{-1}. In general, relatively small values for D were obtained indicating sharp BTC for conservative solutes.

TABLE 38. HYDRODYNAMIC DISPERSION COEFFICIENTS D OF ^3H_2O IN Ap, E1 AND E2 ARREDONDO SOIL HORIZONS

Soil Horizon	Darcy Flow Velocity q	
	$q = 0.5$ cm h^{-1} (cm^2 s^{-1})	$q = 5.0$ cm h^{-1} (cm^2 s^{-1})
Ap	2.5×10^{-4}	8.0×10^{-4}
E1	3.0×10^{-4}	4.5×10^{-4}
E2	5.5×10^{-4}	9.0×10^{-4}

The dispersion coefficients were verified by substituting them into the convective dispersion equation for a conservative or nonreactive solute:

$$\partial C/\partial t = D\partial^2 C/\partial x^2 - \upsilon \partial C/\partial x \qquad [5]$$

and comparing calculated results (Figures 51 and 52) with experimentally determined BTC for tritiated water in the column effluent.

The three horizons of Arredondo fine sand were found to be acidic, therefore, $N_2H_5^+$ was the prevalent form of hydrazine under water-saturated soil conditions. Measured pH values of the soil suspensions for Ap, E1 and E2 Arredondo soil were 4.45, 4.98, and 5.10, respectively.

Titration curves for the Arredondo soil are given in Figure 53. None of the three curves show the characteristic sigmoid shape indicative of a buffered plateau with less buffered regions on either side. The titration curve of the Ap horizon has a smaller slope than that of the E1 horizon, indicating less susceptibility to pH change by increasing amounts of $Ca(OH)_2$. The order of buffering capacity of the three horizons was Ap > E1 > E2 with none of them possessing a strong buffering capacity.

Chemical analysis performed on acid extractions of soil samples (Table 39) indicate a predominance of calcium and aluminium elements. Concentrations of all elements were in the order Ap > E1 > E2 for the three soil horizons.

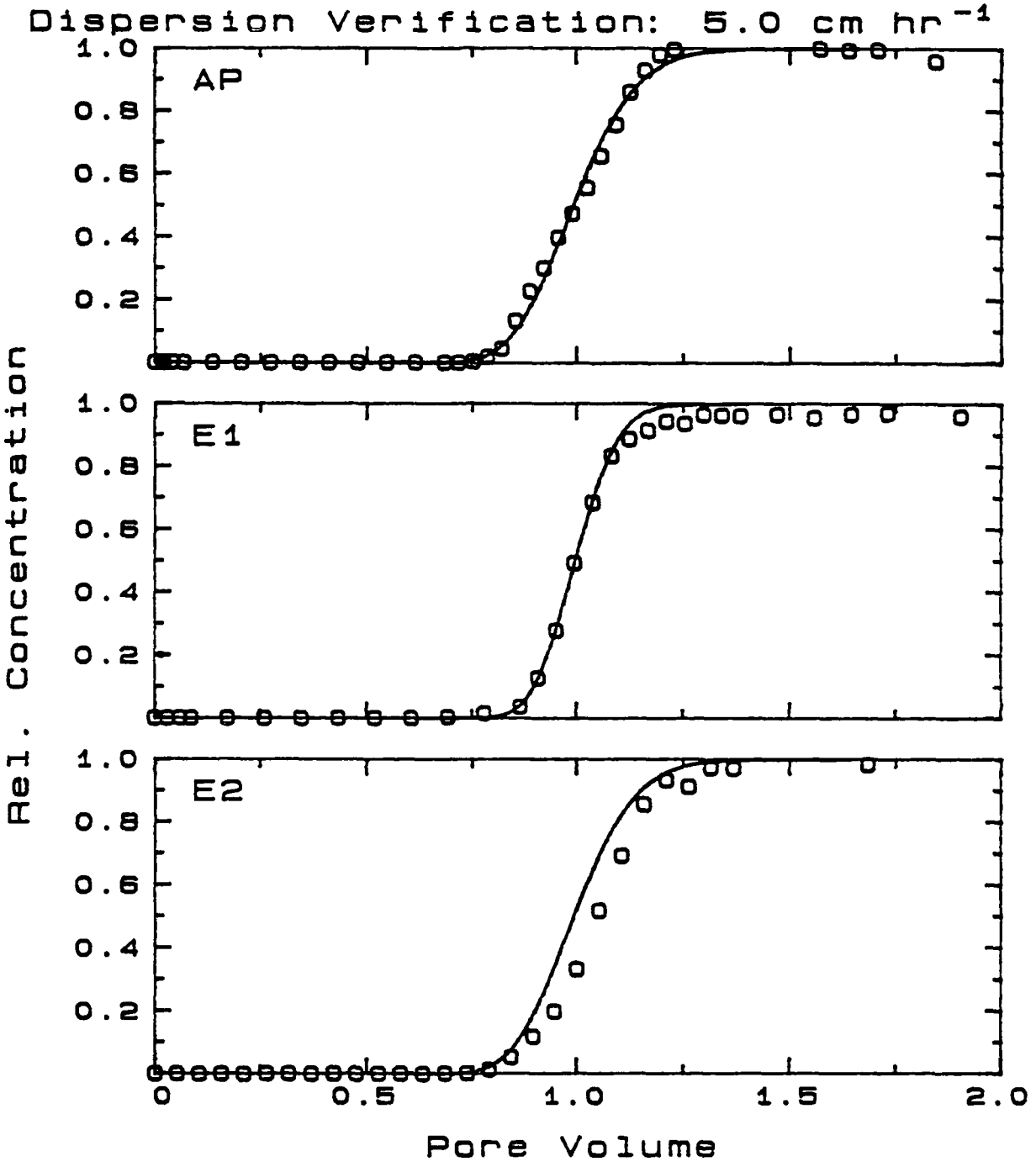


Fig 51. Experimental (Open Circles) and Least-Squares Fit (Smooth Curves) BTC for Tritiated Water in Effluent from Columns of Ap, E1, and E2 Arredondo Soil with Liquid Flux q = 5 cm h^{-1}.

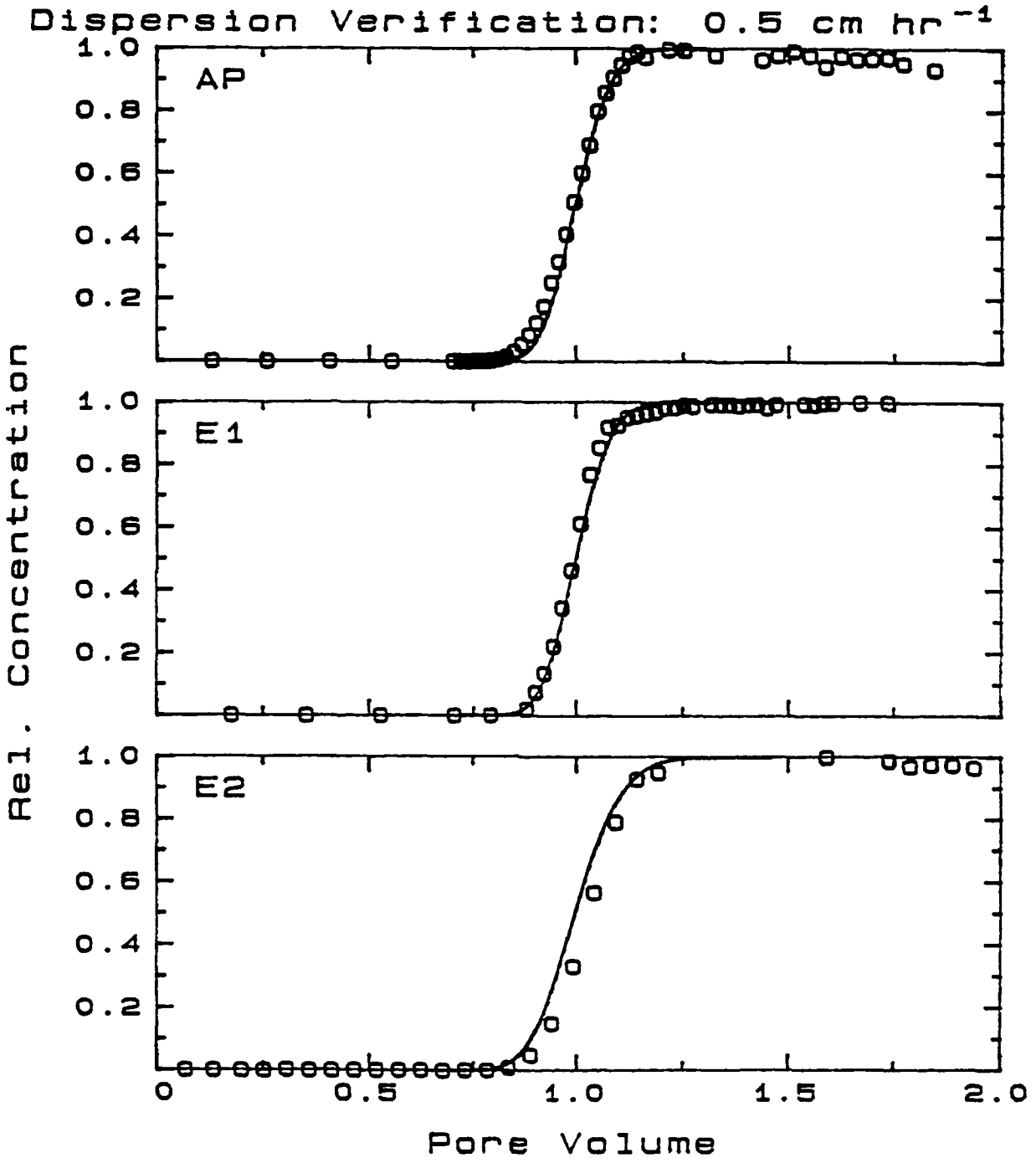


Fig 52. Experimental (Open Circles) and Least-Square Fit (Smooth Curves) BTC for Tritiated Water in Effluent from Columns of Ap, E1, and E2 Arredondo Soil with Liquid Flux q = 0.5 cm h^{-1}.

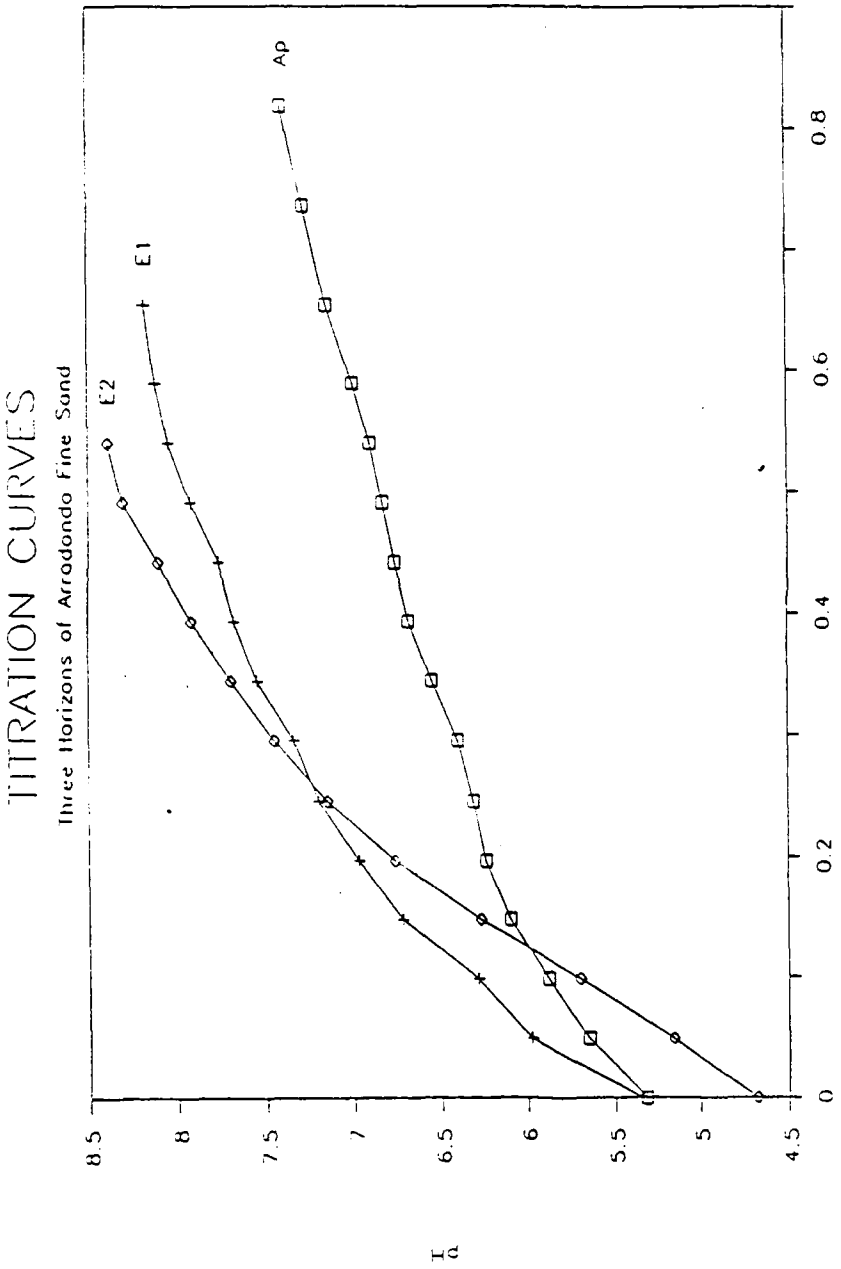


Fig 53. Experimental Titration Curves Over a pH Range from 4.5 to 8.5 for Ap, E1, and E2 Arredondo Soil.

TABLE 39. CHEMICAL ANALYSIS OF ARREDONDO SOIL FROM Ap, E1, AND E2 HORIZONS

Element	Concentration		
	Ap Soil ($g\ Mg^{-1}$)	E1 Soil ($g\ Mg^{-1}$)	E2 Soil ($g\ Mg^{-1}$)
Calcium	561	29	16
Magnesium	24	4	4
Sodium	24	7	8
Potassium	19	7	5
Iron	78	77	46
Aluminum	691	359	140

X-ray analysis of the clay films from Ap, E1, and E2 Arredondo soil revealed peaks (Figure 54) at angles corresponding to the D-spacing of kaolinite. No smectite clays or significant mineral oxides were found.

Descriptive data for soil columns used in miscible displacement experiments is given in Table 40. A total of 32 soil columns were used --- 8 for Ap soil, 12 for E1 soil and 12 for E2 soil. Treatments for each soil included three concentrations, C_o, of hydrazinium (low, medium, and high) in applied influent, two Darcy fluxes (low and high), and two methods for applying the hydrazinium solutions (pulse and continuous). Values for soil bulk density, ρ, and volumetric water contents, θ, for soil columns is given in Table 41. Experimentally-determined pH and concentrations of $N_2H_5^+$ and Ca^{2+} in column effluent are reported in (Figures 68-75 in Appendix) for Ap soil, (Figures 76-83 in Appendix) for E1 soil and (Figures 84-91 in Appendix) for E2 soil. Soil columns designated as 1 through 4 in Table 40 for Ap soil with influent having low concentration of hydrazinium were not used in miscible displacement experiments.

Incomplete recoveries (ratios of quantities of $N_2H_5^+$ removed from the soil in effluent to quantities applied in influent) of hydrazinium (Table 42) in effluent from soil columns indicate that the chemical was undergoing a degree of irreversible removal or chemisorption by soil components as the influent was displaced through the columns. For each soil, recoveries of hydrazinium were less when q was 0.5 cm h^{-1} than when q was 5 cm

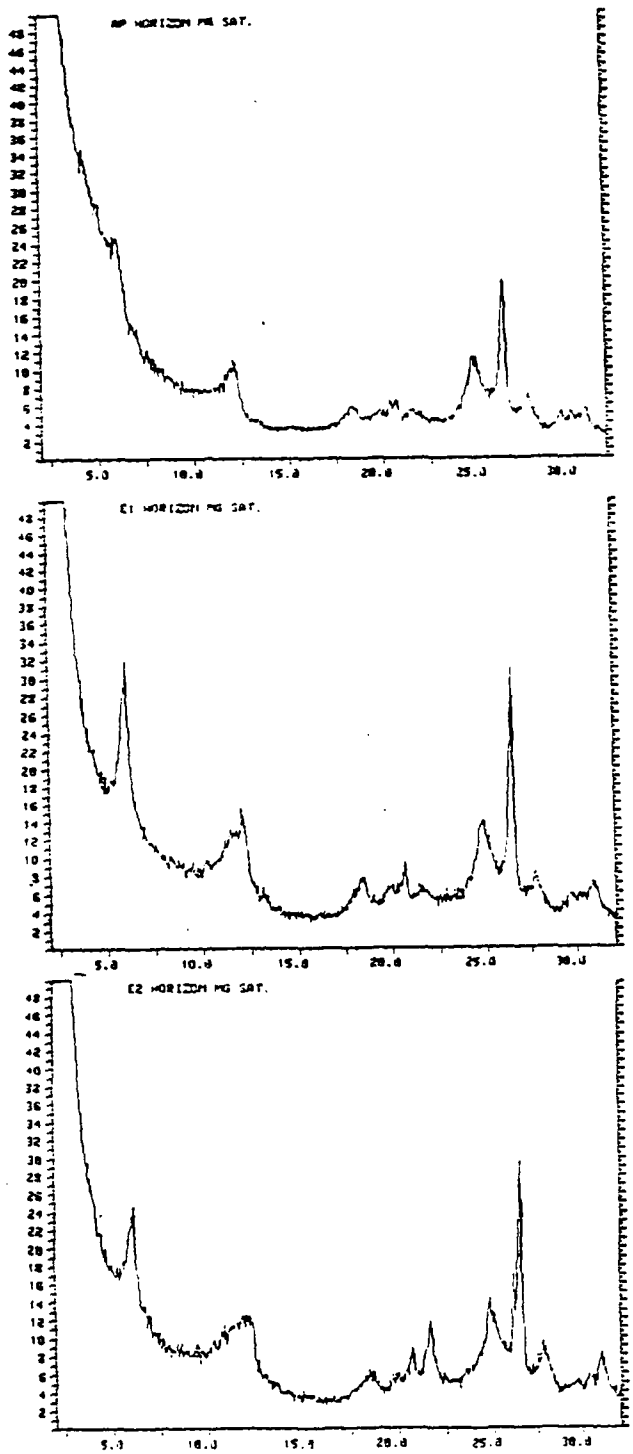


Figure 54. X-ray Diffraction Patterns for Ap, E1, and E2 Arredondo Soil.

TABLE 40. DESCRIPTIVE INFORMATION FOR MISCIBLE DISPLACEMENT OF HYDRAZINIUM INFLUENT THROUGH COLUMNS OF ARREDONDO FINE SAND. VARIABLES INCLUDE 3 SOIL HORIZONS (Ap, E1, AND E2), 3 HYDRAZINIUM CONCENTRATIONS C_o IN APPLIED INFLUENT (L = LOW, M = MEDIUM, AND H = HIGH), TWO VALUES FOR q (L = LOW AND H = HIGH), AND 2 METHODS OF INFLUENT APPLICATION (P = PULSE AND C = CONTINUOUS).

Column Designator	Soil Horizon	Influent C_o (mmol l^{-1})	Flux q (cm h^{-1})	Influent Application Method	Influent Volume of Applied $N_2H_5^+$ Solutions (No. Pore Volumes)	Effluent Volume Collected (No. Pore Volumes)
1	Ap	(L)	0.5 (L)	Pulse	*no data*	*no data*
2	"	(L)	0.5 (L)	Continuous	" "	" "
3	"	(L)	5.0 (H)	Pulse	" "	" "
4	"	(L)	5.0 (H)	Continuous	" "	" "
5	"	2.822 (M)	0.5 (L)	Pulse	2.53	13.91
6	"	2.711 (M)	0.5 (L)	Continuous	23.91	23.91
7	"	2.742 (M)	5.0 (H)	Pulse	2.01	11.66
8	"	3.185 (M)	5.0 (H)	Continuous	20.48	20.48
9	"	10.45 (H)	0.5 (L)	Pulse	2.70	17.82
10	"	10.45 (H)	0.5 (L)	Continuous	26.98	26.98
11	"	8.884 (H)	5.0 (H)	Pulse	1.72	14.05
12	"	10.48 (H)	5.0 (H)	Continuous	10.84	10.84
13	E1	0.1319 (L)	0.5 (L)	Pulse	1.88	14.4
14	"	0.1319 (L)	0.5 (L)	Continuous	13.5	13.5
15	"	0.1304 (L)	5.0 (H)	Pulse	1.88	5.58

TABLE 40. DESCRIPTIVE INFORMATION FOR MISCIBLE DISPLACEMENT OF HYDRAZINIUM INFLUENT THROUGH COLUMNS OF ARREDONDO FINE SAND. VARIABLES INCLUDE 3 SOIL HORIZONS (Ap, E1, AND E2), 3 HYDRAZINIUM CONCENTRATIONS C_o IN APPLIED INFLUENT (L = LOW, M - MEDIUM, AND H = HIGH), TWO VALUES FOR q (L = LOW AND H = HIGH), AND 2 METHODS OF INFLUENT APPLICATION (P = PULSE AND C = CONTINUOUS). (Concluded)

16	"	0.1187	(L)	5.0 (H)	Continuous	11.1	11.1
17	"	2.439	(M)	0.5 (L)	Pulse	1.79	11.9
18	"	2.925	(M)	0.5 (L)	Continuous	10.9	10.9
19	"	2.967	(M)	5.0 (H)	Pulse	1.76	11.5
20	"	2.940	(M)	5.0 (H)	Continuous	11.6	11.6
21	"	10.64	(H)	0.5 (L)	Pulse	1.98	11.3
22	"	8.362	(H)	0.5 (L)	Continuous	12.2	12.2
23	"	12.20	(H)	5.0 (H)	Pulse	2.18	10.4
24	"	10.89	(H)	5.0 (H)	Continuous	11.7	11.7
25	E2	0.1226	(L)	0.5 (L)	Pulse	2.02	12.6
26	"	0.1187	(L)	0.5 (L)	Continuous	10.0	10.00
27	"	0.1319	(L)	5.0 (H)	Pulse	1.90	8.54
28	"	0.1569	(L)	5.0 (H)	Continuous	8.50	8.50
29	"	2.729	(M)	0.5 (L)	Pulse	2.04	12.9
30	"	3.098	(M)	0.5 (L)	Continuous	6.60	6.60
31	"	3.072	(M)	5.0 (H)	Pulse	1.90	12.0
32	"	2.729	(M)	5.0 (H)	Continuous	6.94	6.94
33	"	9.266	(H)	0.5 (L)	Pulse	1.83	10.9
34	"	10.41	(H)	0.5 (L)	Continuous	9.44	9.44
35	"	9.128	(H)	5.0 (H)	Pulse	2.03	12.9
36	"	13.71	(H)	5.0 (H)	Continuous	12.6	12.6

TABLE 41. PERTINENT PARAMETERS FOR MISCIBLE DISPLACEMENT EXPERIMENTS. SOIL PARTICLE DENSITY OF 2.65 Mg m^{-3} WAS ASSUMED IN ORDER TO ESTIMATE POROSITY f USING $f = 100\ (1 - \rho/2.65)$.

Column Designator	Bulk Density ($Mg\ m^{-3}$)	Estimated Porosity f ($m^3\ m^{-3}$)	Soil Water Content ($m^3\ m^{-3}$)	Degree Water Saturation θ/f (percent)	Pore Water Velocity υ ($cm\ h^{-1}$)
1	no data				
2	"	"			
3	"	"			
4	"	"			
5	1.43	0.46	0.36	78	1.39
6	1.42	0.46	0.27	59	1.85
7	1.47	0.55	0.28	51	17.9
8	1.46	0.45	0.30	67	16.7
9	1.44	0.46	0.33	72	1.52
10	1.44	0.46	0.24	52	2.08
11	1.42	0.46	0.30	65	16.7
12	1.46	0.45	0.25	56	20.0
13	1.61	0.39	0.26	67	1.92
14	1.61	0.39	0.22	56	2.27
15	1.56	0.41	0.28	68	17.9
16	1.61	0.39	0.26	67	19.2
17	1.63	0.38	0.26	68	1.92
18	1.59	0.40	0.24	60	2.08
19	1.59	0.40	0.26	65	19.2
20	1.61	0.39	0.23	59	21.7
21	1.62	0.39	0.25	64	2.00
22	1.60	0.39	0.24	62	2.08
23	1.59	0.40	0.27	68	18.5
24	1.61	0.39	0.26	67	19.2
25	1.64	0.38	0.26	68	1.92
26	1.60	0.40	0.25	63	2.00

TABLE 41. PERTINENT PARAMETERS FOR MISCIBLE DISPLACEMENT EXPERIMENTS. SOIL PARTICLE DENSITY OF 2.65 Mg m^{-3} WAS ASSUMED IN ORDER TO ESTIMATE POROSITY f USING F = 100 (1-p/2.65).(concluded)

27	1.63	0.38	0.24	63	20.8
28	1.63	0.38	0.25	66	20.0
29	1.65	0.38	0.23	61	2.17
30	1.61	0.39	0.26	67	1.92
31	1.65	0.38	0.24	63	20.8
32	1.63	0.38	0.25	66	20.0
33	1.62	0.39	0.27	69	1.85
34	1.63	0.38	0.25	66	2.00
35	1.63	0.38	0.25	66	20.00
36	1.63	0.38	0.25	66	20.00

TABLE 42. RELATIVE RECOVERIES (100 TIMES THE RATIOS OF QUANTITIES OF SOLUTE APPLIED IN INFLUENT AND QUANTITIES ELUTED IN EFFLUENT) OF HYDRAZINIUM IN SOIL COLUMN EFFLUENT. THREE LETTERS ARE USED AS TREATMENT DESIGNATORS FOR SOIL COLUMNS. THE FIRST LETTER IS THE CONCENTRATION, C_o, OF INFLUENT (L = LOW, M = MEDIUM, AND H = HIGH), THE SECOND LETTER IS LIQUID FLUX q (L = LOW AND H = HIGH), AND THE THIRD LETTER IS THE METHOD OF INFLUENT APPLICATION (P = PULSE AND C = CONTINUOUS).

Column Designator	Soil Horizon	Treatment Designator	Recovery Hydrazinium (percent)
1	Ap	L L P	no data
2	"	L L C	"
3	"	L H P	"
4	"	L H C	"
5	"	M L P	0
6	"	M L C	38.2
7	"	M H P	0
8	"	M H C	59.3
9	"	H L P	17.9

TABLE 42. RELATIVE RECOVERIES (100 TIMES THE RATIOS OF QUANTITIES OF SOLUTE APPLIED IN INFLUENT AND QUANTITIES ELUTED IN EFFLUENT) OF HYDRAZINIUM IN SOIL COLOMN EFFLUENT. THREE LETTERS ARE USED AS TREATMENT DESIGNATORS FOR SOIL COLUMNS. THE FIRST LETTER IS THE CONCENTRATTION, C_0, OF INFLUENT (L = LOW, M = MEDIUM, AND H = HIGH), THE SECOND LETTER IS LIQUID FLUX q (L = LOW AND H = HIGH), AND THE THIRD LETTER IS THE MEHTOD OF INFLUENT APPLICATION (P = PULSE AND C = CONTINUOUS).(Concluded)

10	"	H L C	38.2
11	"	H H P	51.8
12	"	H H C	48.9
13	E1	L L P	0
14	"	L L C	0
15	"	L H P	0
16	"	L H C	0.9
17	"	M L P	31.7
18	"	M L C	58.7
19	"	M H P	63.7
20	"	M H C	88.1
21	"	H L P	76.0
22	"	H L C	95.6
23	"	H H P	84.4
24	"	H H C	92.3
25	E2	L L P	0
26	"	L L C	31.2
27	"	L H P	40.2
28	"	L H C	44.0
29	"	M L P	79.7
30	"	M L C	75.9
31	"	M H P	79.1
32	"	M H C	79.2
33	"	H L P	92.6
34	"	H L C	92.5
35	"	H H P	114.6
36	"	H H C	81.6

h^{-1}. This result was expected since the smaller liquid flux gives a ten-fold greater residence time in the column compared to the larger flux and thus permits greater opportunity for chemisorption to occur. This result

strongly implies that the irreversible chemisorption behaved as a kinetic reaction. The order of hydrazinium recoveries in column effluent was generally in the order Ap < E1 < E2 which was related to the relative contents of organic carbon in these soil horizons.

For each soil, recoveries of hydrazinium in column effluent tended to be the highest when the high concentration influent was applied. When influent contained high hydrazinium concentrations, recoveries ranged from 18 to 52 percent for Ap soil, 76 to 96 percent for E1 soil, and 82 to 100 percent for E2 soil. When influent contained medium hydrazinium concentrations, recoveries ranged from 0 to 59 percent for Ap soil, 32 to 88 percent for E1 soil, and 76 to 80 percent for E2 soil. Recoveries ranged from 0 to 0.9 percent for E1 soil when influent contained low hydrazinium concentration and ranged from 0 to 44 percent for E2 soil. Thus irreversible chemisorption appeared to be inversely related to the concentration of hydrazinium applied as influent.

Hydrazinium recovery in column effluent also was generally greater for conditions where hydrazinium solutions were applied continuously for a large number of pore volumes of effluent versus conditions where hydrazinium solutions were applied as a wide pulse for a relatively small number of pore volumes of effluent. Acidified solutions of 0.001 N $CaCl_2$ solutions without any $N_2H_5^+$ were applied with a pump to each column after a pulse of N_2H_5 solution had been applied. Recoveries observed by pulse and continuous applications of medium concentration influent with high liquid flux were 0 and 59 percent, respectively for Ap soil, 64 and 88 percent for E1 soil, and 79.1 and 79.2 percent for E2 soil.

Breakthrough data for hydrazinium in column effluent (Figures 68-91 in Appendix) showed that $N_2H_5^+$ cations were most mobile in the E2 soil, less mobile in the E1 soil and considerably less mobile in the Ap surface soil. Hydrazinium BTC for E1 and E2 soil columns that received high concentration influent was characterized by early appearance of $N_2H_5^+$ cations in the effluent, sharp initial portions of the curve, and relatively high maximum concentrations. In contrast, hydrazinium BTC for Ap soil columns that received high concentration influent was characterized by delayed appearance of $N_2H_5^+$ in the effluent, more diffuse initial portions of the curve, and lower maximum concentrations. The general

order of hydrazinium mobility in each soil for different concentrations, C_o, of hydrazinium in applied influent was: high C_o > medium C_o > low C_o. As expected for kinetically-driven sorption processes, the mobility of applied $N_2H_5^+$ was enhanced in each soil by increasing Darcy liquid flux from 0.5 to 5.0 cm h^{-1}.

BTC for Ca^{2+} in effluent from E2 soil (Figures 84-91 in Appendix) were obviously related to hydrazinium BTC and indicate that some of the exchangeable Ca^{2+} ions were replaced by $N_2H_5^+$ cations by cation exchange as hydrazinium influent was displaced through the columns. When high concentration influent was applied continuously, the breakthrough of $N_2H_5^+$ in the effluent was preceded by a peak in the Ca^{2+} BTC. When influent was applied as a pulse, the Ca^{2+} BTC was characterized with a peak prior to hydrazinium breakthrough as well as a trough with a minimal concentration that occurred after the hydrazinium breakthrough. The Ca^{2+} peak was associated with exchangeable Ca^{2+} cations being replaced with $N_2H_5^+$ cations and the Ca^{2+} breakthrough was associated with exchangeable $N_2H_5^+$ cations being replaced with Ca^{+2} ions. The magnitude of the peaks and breakthroughs for Ca^{+2} BTC were much less detectable for effluent solutions with medium and low concentrations of $N_2H_5^+$.

When hydrazinium influent was applied continuously to soil columns, a rather sharp decrease in pH occurred as $N_2H_5^+$ cations first appeared in the effluent. For pulse applications of influent, pH also increased sharply as hydrazinium concentration in the effluent sharply decreased. The decrease in pH (increase in hydrogen ion activity) was attributed to exchangeable H^+ ions being replaced with $N_2H_5^+$ cations in the incoming influent. The increase in pH was attributed to exchangeable $N_2H_5^+$ being replaced with H^+ ions.

Aqueous acidic 0.01N $CaCl_2$ solutions with specificed hydrazinium concentration, C_o, were miscibly displaced through 30-cm long columns of water-saturated Arredondo fine sand under conditions of steady liquid flux in order to assess the influence of pore water velocity v, C_o of input hydrazinium solutions, and two methods of application of input solution upon the transport of hydrazinium. Darcy flow velocities of approximately 0.5 and 5.0 cm h^{-1} corresponded roughly to pore velocities of 2 and 20 cm h^{-1} where soil water content was about 0.25 cm^3cm^{-3}. Pore

velocities of 2 and 20 cm h^{-1} provided residence times of about 15 and 1.5 hours, respectively, within a 30-cm length soil column. Low (C_o = 6.6 mg l^{-1}), medium (C_o = 165 mg l^{-1}) and high (C_o = 660 mg l^{-1}) hydrazinium concentrations were applied as input solutions. Two methods of application were used: continuous (a step-function increase from zero concentration in input solution to C_o) and pulse. Columns were hand-packed with soil material from Ap, E1, and E2 horizons from a profile of Arredondo fine sand. Although the hand-packed columns represent physically-modified soil with respect to soil structure, their use provided a convenient means to investigate the fate and transport of hydrazinium in cylindrical bulk volumes of chemically and physically homogenous soil taken from the three profile horizons of primary interest.

Experimental BTC for hydrazinium in effluent from soil columns of Arredondo fine sand revealed that mobility of applied hydrazinium increased with increasing pore water velocity and with increasing C_o in applied solution. Similar behavior typically occurs with other reactive chemicals such as orthophosphate (references 50, 51, 59) when applied to sandy soils.

Relative mobility of hydrazinium applied as influent to columns of water-saturated Arredondo soil differed with soil taken from different horizons in the profile. Mobility of applied hydrazinium increased with increasing depth of the soil horizons. Minimal mobility occurred in columns of Ap surface soil which had the highest contents of organic matter and clay minerals and was maximal in columns of E2 subsoil. Hydrazinium mobility was intermediate for the E1 subsoil horizon.

For hydrazinium BTC from columns of all three soil profile horizons which received influent solutions with the highest C_o (660 mg l^{-1}), the occurrence of early breakthrough (less than 2 pore volumes) in the effluent as well as steep curves immediately following initial breakthrough indicated high mobility of the solute during transport. Considerable tailing of BTC's for both pulse- and continuous-applications of influent indicated that kinetic reaction mechanisms were operative in controlling the hydrazinium concentration in the solution phase of the soil. Incomplete recovery of applied hydrazinium in the effluent revealed that some

of the solute was removed irreversibly during miscible displacement of the influent.

Fractional recoveries of applied hydrazinium ions in effluent from columns of E1, E2 and Ap soil materials were especially low when the influent had the low hydrazinium concentration (6.6 mg l^{-1}). Influent pulses with low C_o resulted in no elution of $N_2H_5^+$ ions from columns of Ap soil. Low recoveries also occurred when medium concentration (165 mg l^{-1}) influent was applied to the Ap topsoil. Thus, quantities of hydrazinium irreversibly removed from the soil solution during displacement of influent through soil columns were directly related to contents of organic matter and clay minerals in the soil matrix. For all soil horizons and all C_o in influent, tenfold increases in pore water velocity from approximately 2 to 20 cm h^{-1} generally increased fractional recoveries of $N_2H_5^+$ and gave initially steeper slopes in the BTC. This observation indicates that the shorter solute residence time provided less time for the kinetic reactions between hydrazinium and soil components.

Breakthrough curves for Ca^{2+} and H^+ ions in effluent for each column were observed to be related to hydrazinium BTC. When influent with high C_o was applied to soil columns, initial breakthrough of $N_2H_5^+$ in the effluent was preceded by an increase in H^+ concentration (i.e., a decrease in pH) which in turn was preceded by an increase in Ca^{2+} concentration. This observation implies heavily that the BTC's for these three cation species were at least partially related by the physical process of ion exchange. A further implication is that soil exchange sites exhibited the following order of preference for the 3 ion species: $H^+ > Ca^{2+} > N_2H_5^+$. Equivalent fractions of Ca^{2+} ion species in the soil solution initially (Table 43) exceeded 99.6 percent whereas it was only 33.3 percent in the displacing influent which contained 66.6 percent equivalent fraction of $N_2H_5^+$ ions for the case of high C_o. Thus the ratio of equivalent fractions of $N_2H_5^+$ and of Ca^{2+} in the influent was approximately 2.0 which provided a mass-action effect that permitted the applied $N_2H_5^+$ ions in solution to competitively exchange with Ca^{2+} and H^+ ions initially present on the soil exchange sites. Recent work by Rhue and Mansell (reference 56) has shown that H^+ ions can provide important ternary effects upon binary exchange between Na^+ and Ca^{2+} as well as K^+

TABLE 43. EQUIVALENT FRACTIONS OF Ca^{2+}, $N_2H_5^+$, AND H^+ ION SPECIES IN COLUMN INFLUENT WITH CONCENTRATIONS C_o OF 0, 6.6, 165, AND 660 mg l^{-1}. ALL SOLUTIONS CONTAINED 0.01 N $CaCl_2$ AND pH WAS ADJUSTED TO 4.5. CONCENTRATIONS RATHER THAN ACTIVITIES WERE USED IN CALCULATIONS.

Influent Concentration of $N_2H_5^+$ (mg l^{-1})	Influent Molarity of N_2H_5 (mmol(+) cm^{-3})	Equivalent Fraction of Ion Species in Influent		
		Ca^{2+}	$N_2H_5^+$	H^+
		percent	percent	percent
0	1.033×10^{-2}	99.694	0	0.306
6.6	1.023×10^{-2}	97.762	1.955	0.309
165	1.503×10^{-2}	66.534	33.267	0.210
660	3.003×10^{-2}	33.300	66.600	0.105

and Ca^{2+} ion species in acid soil systems. In Cecil soil, pH-dependent charge sites in the exchange complex were shown to have a greater affinity for divalent Ca than for monovalent Na^+ or K^+. Ca ions were therefore expected to displace more exchangeable H^+ ions than would Na^+ or K^+. A mathematical model for describing ion transport and multicomponent ion-exchange in soil columns was recently modified (reference 52) to permit the use of variable exchange selectivity coefficients. The modified model was observed to simulate a ternary ion system with Ca^{2+}, Mg^{2+} and Na^+.

When influent solutions with low and medium concentrations of $N_2H_5^+$ were applied to soil columns, BTC's for Ca^{2+} and H^+ showed increasing concentrations of these species prior to $N_2H_5^+$ BTC but the magnitudes of these increases were much less than that for high C_o. This effect was expected since the ratios of equivalent fractions of $N_2H_5^+$ and Ca^{2+} ion species in the low and medium concentration influent solutions were only 0.199 and 0.500, respectively. Thus, $N_2H_5^+$ would be expected to be much less competitive for exchange with Ca^{2+} and H^+ ions present in the soil

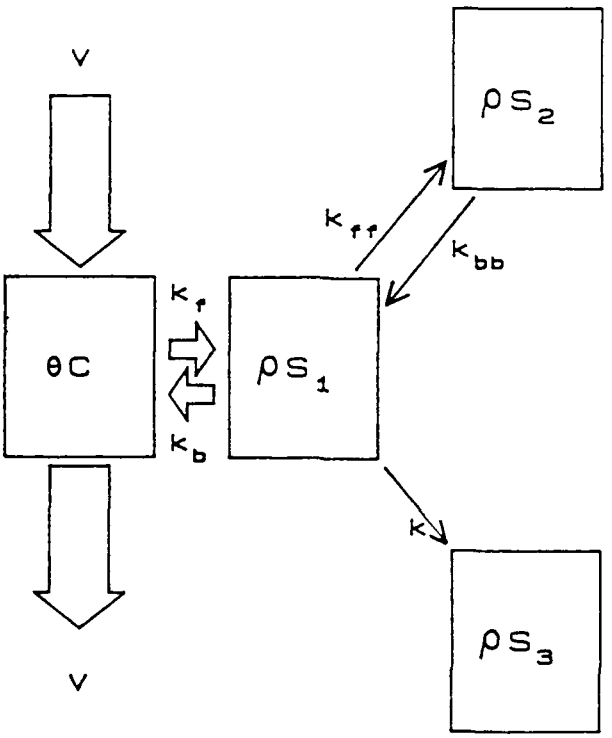
exchange complex for these influents relative to that for influent with high C_o where the ratio of equivalent fractions of $N_2H_5^+$ and Ca^{2+} was 2.00.

Hydrazinium BTC presented in this report as well as by Downs et al (reference 45) and Mansell et al (reference 52) provide important first-time experimental data that shows that $N_2H_5^+$ applied to columns of acid sandy Arredondo soil was in fact transported through the soil under laboratory conditions of steady flow and water-saturation of the porous media. The relatively high mobility of this toxic chemical for conditions of high pore water velocity and high concentration of $N_2H_5^+$ of applied influent have significant implications with respect to potential contamination of groundwater in the event of accidental spillage or leakage of hydrazinium into Arredondo fine sand soil.

C. MATHEMATICAL MODEL FOR ONE-DIMENSIONAL TRANSPORT OF HYDRAZINIUM IN SOIL

1. Model Description

A simple mathematical model was developed to describe the convective-dispersive transport and reactions of $N_2H_5^+$ during steady liquid flow through columns of water-saturated Arredondo fine sand. Acid conditions were assumed so that only protonated ionic forms of hydrazine are considered. Hydrazinium ions in the soil solution are assumed to be subject to relatively fast kinetic physical sorption (i.e., ion exchange). Physically sorbed $N_2H_5^+$ ions are simultaneously subject to irreversible chemisorption and to relatively slow kinetic non specific sorption (i.e., tightly-bound sorption). A schematic diagram for hydrazinium reactions assumed in the model is given as follows:



$k_f > k_{ff}$ and $k_b > k_{bb}$

where C is the concentration of $N_2H_5^+$ (mol l^{-1}) in the mobile solution phase, θ is the volumetric fraction of bulk soil (cm^3cm^{-3}) occupied with liquid, υ is the average pore-water velocity (m s^{-1}) of the soil solution, S_1 is the concentration (mol kg^{-1}) of hydrazinium in the physically-sorbed or exchangeable form, S_2 is the concentration (mol kg^{-1}) of hydrazinium non-specifically sorbed or in the tightly-bound form, S_3 is the concentration (mol kg^{-1}) of hydrazinium in the chemisorbed form, ρ is the dry bulk density (Mg m^{-3}) of the soil, k_f and k_b are forward and backward rate coefficients (s^{-1}) for physical sorption, k_{ff} and k_{bb} are forward and backward rate coefficients for kinetic non-specific sorption, and K is the rate coefficient for irreversible chemisorption confined with biodegradation. For steady flow in a soil profile with different horizons, the parameters θ, ρ, k_f, k_b, k_{ff}, k_{bb}, and k would vary with depth due to differences in properties of soil in different horizons. For steady flow in the soil columns these parameters were assumed to be constant for each soil horizon. The sum of S_1, S_2 and S_3 is designated as S the total concentration (mol kg^{-1}) of sorbed $N_2H_5^+$ for a given location in the soil. Although S may change with time and depth, it is important to remember that sorbed hydrazinium ions are essentially immobile. The rate of chemisorption is assumed to be directly porportional to S_1 or the concentration of hydrazinium in the physically-sorbed form but is also limited by S_{3m} or the maximum concentration in the chemisorbed form

$$\partial S_3/\partial t = +k[1-S_3/S_{3m}]S_1 \qquad [6]$$

where t is time (s). Disappearance of $N_2H_5^+$ by the mechanism of microbial degradation was experimentally shown to be a relatively small effect for the short time durations for each of the soil column experiments and was thus omitted from the model formulation. The parameter k was, thus assumed to provide the rate coefficient for only chemisorption.

The model requires that two equations be solved numerically, one for convective-dispersive transport

$$\partial C/\partial t = D\ \partial^2 C/\partial z^2 - \upsilon\partial C/\partial z + [\rho/\theta]k_b\ S_1 - k_f C \qquad [7]$$

and one for the net rate of sorption of $N_2H_5^+$

$$\partial S_1/\partial t = [\theta/\rho]k_f C + k_{bb} S_2 - [k_b + k_{ff} + k(1-S_3/S_{3m})]S_1 \qquad [8]$$

where D is the hydrodynamic dispersion coefficient ($m^2 s^{-1}$) and z is vertical distance in a soil column of length L. Steady liquid flow (i.e., υ is constant) was maintained at all times and dilute aqueous solutions with concentration C_o was applied either continuously (i.e., as a step-function input) or as a wide pulse to the soil surface (z = o) using a flux boundary condition

$$\upsilon C_o = \upsilon C - D\partial C/\partial z \qquad [9]$$

for $t > 0$.

The boundary condition imposed at the bottom (z = L) of the column was

$$\partial C/\partial z = 0 \qquad [10]$$

for $t > 0$.

Initial conditions were such that values for C, S_1, S_2, and S_3 were zero in the soil prior to application of aqueous solutions of hydrazinium.

A Crank-Nicholson finite differencing technique (reference 59) was used to numerically solve equations [7] and [8] subject to boundary conditions [9] and [10]. The distance between nodes in space were kept constant as Δz = 0.25 cm and time steps Δt were adjusted between 5 and 300 s in order to insure minimal cumulative mass balance errors in numerical simulations for C(z,t).

2. Experimental Data Used in Simulations

Air-dry soil from Ap (topsoil), E1 (subsoil), and E2 (subsoil) horizons of a profile of Arredondo fine sand were carefully hand-packed into glass chromatographic columns with 30-cm length and 5.0-cm inside diameter. The properties of the three horizons (Table 44) were such that although the content of sand-sized particles exceeded 90 percent in each,

the clay content of the topsoil was 1.5 times greater than that in the E2 subsoil horizon. The organic carbon content for the Ap topsoil was 13.1 times greater than that in the E2 subsoil horizon but the organic matter content in the E1 subsoil horizon was only 2.4 times greater than in the E2 horizon. Since clay minerals and organic matter are typically chemically reactive components of soils, one might expect sorption of $N_2H_5^+$ to be greater in the Ap horizon and least in the E2 horizon. For the same reasons, $N_2H_5^+$ mobility should be greater in E2 soil material and least in Ap soil. Miscible displacement of hydrazinium solutions through columns of these soils showed the mobility of $N_2H_5^+$ to be in the order Ap < E1 < E2.

TABLE 44. PROPERTIES OF ARREDONDO FINE SAND FOR Ap, E1, and E2 HORIZONS.

Horizon	Depth in Profile (cm)	Clay Content (percent)	Silt Content (percent)	Sand Content (percent)	Organic Carbon Content (percent)
Ap	0-20	2.6	7.3	90.1	1.84
E1	20-80	1.7	4.9	93.4	0.34
E2	80-150	1.8	3.7	94.5	0.14

Acidic aqueous (pH 4.5) 0.01 N Ca Cl_2 solution was used to wet each soil column. The solution was pumped at a constant Darcy flow velocity ($q = v\theta$) for a period of 24 hours to establish steady liquid flow and to saturate soil exchange sites with Ca^{2+} ions. Oxygen was purged from the solution using helium gas prior to application to the soil columns. Two Darcy flow velocities q, 5×10^{-4} and 50×10^{-4} cm s^{-1}, and three concentrations, C_o, in input solutions, 6.6 (low), 165 (medium), and 660 (high) mg l^{-1}, were used for the column experiments. Input or influent $CaCl_2$ solutions with specified C_o were applied to columns by two methods, continuously as a step-function and as a 2-pore volume pulse followed by $CaCl_2$ solution without hydrazinium (C_0=0). Effluent aliquots from each column were chemically analyzed (reference 62) for $N_2H_5^+$ until a sufficient number of pore volumes of effluent had been collected so that concentrations of $N_2H_5^+$ were constant. BTC for hydrazinium (plots of C/C_o

versus the number of pore volumes p) were then used to evaluate model simulations. Parameters for columns used in model simulations are given in Table 45.

TABLE 45. PARAMETERS FOR COLUMNS OF ARREDONDO FINE SAND THAT RECEIVED APPLICATIONS OF HYDRAZINIUM SOLUTIONS.

Soil Horizon	Column Number	C_o Input Concentration*	Application Method	D Dispersion Coefficient (10^{-4} cm^2 s^{-1})	υ Pore Velocity (10^{-4} cm s^{-1})
Ap	1	Medium	Pulse	2.694	49.6
	2	Medium	Pulse	1.556	3.9
	3	Medium	Continuous	2.694	46.3
	4	Medium	Continuous	1.556	5.1
	5	High	Pulse	2.694	46.3
	6	High	Pulse	1.556	4.2
	7	High	Continuous	2.694	55.6
	8	High	Continuous	1.556	5.8
E1	1	Medium	Pulse	2.694	53.4
	2	Medium	Pulse	1.556	5.3
	3	Medium	Continuous	2.694	60.4
	4	Medium	Continuous	1.556	5.8
	5	High	Pulse	2.694	51.4
	6	High	Pulse	1.556	5.6
	7	High	Continuous	2.694	53.4
	8	High	Continuous	1.556	5.8
E2	1	Medium	Pulse	2.694	57.9
	2	Medium	Pulse	1.556	6.0
	3	Medium	Continuous	2.694	55.6
	4	Medium	Continuous	1.556	5.3
	5	High	Pulse	2.694	55.6
	6	High	Pulse	1.556	5.1
	7	High	Continuous	2.694	55.6
	8	High	Continuous	1.556	5.8

*Medium and high concentrations were approximately 165 and 660 mg l^{-1} of hydrazinium, respectively.

Optimized values of reaction rate coefficients for E1 and E2 subsoil horizons were obtained by calibrating, (i.e., best-fitting) the model to BTC's corresponding to high pore water velocity and medium concentration of hydrazinium in the input solution. The calibration was performed by

using sensitivity analysis for selected model input parameters to obtain optimum values for these parameters (Table 46).

3. Results and Discussion of Simulations

TABLE 46. OPTIMIZED VALUES FOR RATE COEFFICIENTS AND OTHER INPUT PARAMETERS USED IN MODEL SIMULATIONS OF HYDRAZINIUM TRANSPORT IN COLUMNS OF E1 and E2 SUBSOIL MATERIALS.

Soil Horizon	Darcy Flux	Reaction Rate Coefficients					S_{3m}
		k_f	k_b	k_{ff}	k_{bb}	k	
	($cm\ s^{-1}$)	(s^{-1})					($mmol\ kg^{-1}$)
E1	$50x10^{-4}$	$9.5x10^{-3}$	$1.0x10^{-2}$	$3.0x10^{-4}$	$1.2x10^{-3}$	$1.25x10^{-5}$	1.738
E1	$5x10^{-4}$	$9.5x10^{-4}$	$1.0x10^{-3}$	$3.0x10^{-5}$	$1.2x10^{-4}$	$1.25x10^{-6}$	1.738
E2	$50x10^{-4}$	$6.00x10^{-3}$	$1.0x10^{-2}$	$3.0x10^{-4}$	$1.2x10^{-3}$	$2.5x10^{-5}$	0.816
E2	$5x10^{-4}$	$6.00x10^{-4}$	$1.0x10^{-3}$	$3.0x10^{-5}$	$1.2x10^{-4}$	$2.5x10^{-6}$	0.816

The optimized rate coefficient values obtained by calibration of hydrazinium BTC for the case of high Darcy flow velocity and medium C_o for columns of E1 and E2 subsoil materials were divided by 10, the ratio of high to low fluxes, to obtain estimates for the case of low Darcy flux and medium C_o in the input solution. This numerical adjustment of rate coefficients for the low water flux was based on the assumption that reaction rate coefficients were inversely porportional to residence times for water molecules in the 30-cm long soil columns. The optimized rate coefficients obtained in this manner indicated a relatively high mobility for hydrazinium applied as solutions with medium C_o to columns of the E1 and E2 soils since the ratios for forward and backward rate coefficients for relatively fast physical sorption k_f/k_b as well as for relatively slow non specific sorption k_{ff}/k_{bb} were less than unity. In addition the magnitudes for the rate coefficients were assumed to be such that $k_f >> k_{ff}$ and $k_b >> k_{bb}$ indicating that physical sorption (i.e., ion exchange) occurred at a faster rate than did non specific sorption. The rate coefficients shown in Table 46 show that the consecutive sorption reactions that control the concentration of hydrazinium in the mobile solution

phase of the Arredondo soil operated in an overall kinetic mode with reversible as well as irreversible behavior.

Values for the maximum concentrations, S_{3m}, of chemisorbed hydrazinium which E1 and E2 soil materials can support (see Table 46) were obtained by using hydrazinium BTC from soil columns which received continuous application of aqueous solution. The difference between the quantities of hydrazinium added as influent solution and hydrazinium eluted in column effluent was assumed to approximate the quantity of hydrazinium irreversibly retained by the soil due to chemisorption. The estimate of S_{3m} for E1 soil was approximately twice that for the E2 soil. Thus, S_{3m} was roughly in proportion to the organic carbon content of the soil. For each soil material S_{3m} was assumed to be invariant with respect to the pore water velocity. Using optimized rate coefficients k_f, k_b, k_{ff}, and k_{bb}, sensitivity analysis of the model for E2 soil columns which received high flux applications of solutions with medium C_o revealed that doubling the magnitude of the rate coefficient k for irreversible chemisorption (Figure 55) tended to decrease the maximum hydrazinium concentrations C/C_o observed in column effluent. The value of $k = 2.50 \times 10^{-5}\ s^{-1}$ provided an optimum value for describing the BTC. Using the optimum k value, sensitivity analysis of the model (Figure 55) for the S_{3m} parameter gave an optimum value of 8.16×10^{-5} mol g^{-1}. A tenfold larger value for S_{3m} resulted in overestimation of maximum C/C_o values for the BTC. The model adequately described the left-hand side of the BTC but overestimated C/C_o for the right-hand side resulting in a larger simulated recovery of $N_2H_5^+$ in the column effluent than actually observed experimentally.

Under conditions with no chemisorption sink ($k = 0$) and no nonspecific sorption ($k_{ff} = k_{bb} = 0$) sensitivity analyses was performed for the BTC for a column of E2 soil with high flux and medium C_o in the influent (Figure 56) using ratios of physical sorption rate coefficients k_f/k_b of 0.3, 0.6, and 1.2. The optimum ratio was shown to be 0.6 in order to adequately simulate the breakthrough of $N_2H_5^+$ in the effluent. The simulated BTC using the ratio of 0.6 however provided much steeper slope than experimentally observed. A ratio of 1.2 tended to overestimate retardation of $N_2H_5^+$ transport and 0.3 underestimated retardation.

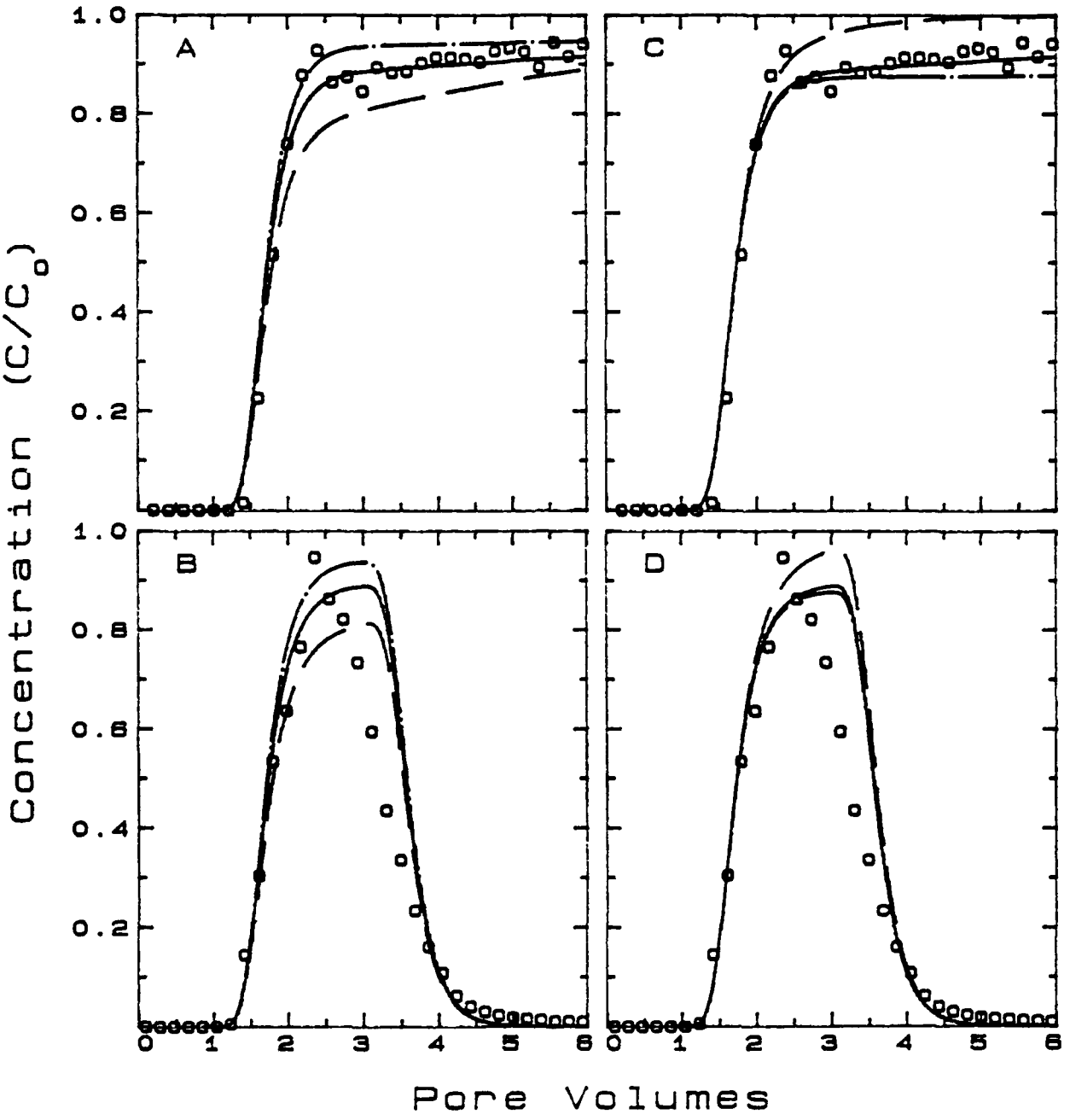


Figure 55. Simulated and Experimental Hydrazinium BTC are Given for Effluent from Columns of E2 Soil Under Conditions of High Flow Velocity q and Medium Hydrazinium Concentration C_o for Input Solution. In Sections A and B Simulated BTC's are Shown for k_5 Values of 1.25×10^{-5} s^{-1} (Dotted Broken Line), 2.50×10^{-5} s^{-1} (Smooth Lines), and 5.0×10^{-5} s^{-1} (Broken Lines). In Sections C and D Simulated BTC's are Presented for S_{3m} Values of 8.16 (Broken Lines), 0.816 (Smooth Lines) and 0.0816 (Dotted Broken Lines) mmol kg^{-1}. Discrete Data Points are Given for Cases Where Hydrazinium Solutions Were Applied Continuously (A and C) and as a Wide Pulse (B and D).

Using the optimum ratio k_f/k_b = 0.6, assuming nonspecific sorption to occur, and assuming no chemisorption sink term (k = 0), further sensitivity analyses (Figure 56) was performed by using ratios of rate coefficients k_{ff}/k_{bb} of 0.125, 0.250, and 0.500 in the model. Increasing this ratio was observed to decrease the slope of the BTC. The value of 0.250 was chosen as the optimum ratio.

Calibrated BTC for E2 soil with medium C_o and high υ are shown in Figure 56 for pulse and continuous applications of influent. Using tenfold smaller values for each of the 5 rate coefficients for E2 soil provided good simulations (Figure 56) for the low pore velocity υ. Using these optimized rate coefficients and parameters for E2 soil, hydrazinium BTC were simulated for cases where high C_o (Figure 57) and low C_o (Figure 58) influent solutions were applied to the soil columns. For high C_o influent the simulated BTC gave reasonable estimation of general shapes; however for pulse applications hydrazinium retardation was slightly over estimated. For low C_o influent the simulated BTC were totally unacceptable. In that case maximum values of C/C_o were overestimated for pulse and continuous applications of influent. Initial slopes of simulated BTC were much steeper than experimental data indicating severe overestimation of hydrazinum mobility, and hydrazinium breakthrough in column effluent was greatly underestimated. Experimental BTC showed extensive irreversible removal of hydrazinium from the applied influent during flow through the soil columns. Obviously solute mobility was much greater experimentally than simulated when low C_o influent was applied to columns of E2 soil.

Calibrated BTC for E1 soil with medium C_o and high υ are shown in (Figure 56) for pulse and continuous applications of influent. Simulations provided good descriptions of experiments for continuous application of influent (Figure 59) but overestimated the area under the BTC for pulse application (Figure 59). For the case of low υ, simulations overestimated maximum values for C/C_o, underestimated retardation, and underestimated irreversible removal of $N_2H_5^+$ from solution. Results were basically the same when simulations were performed for high C_o influent (Figure 60). Simulated BTC for low C_o influent (Figure 61) indicated rather rapid transport of $N_2H_5^+$ through the E1 soil but experimental results

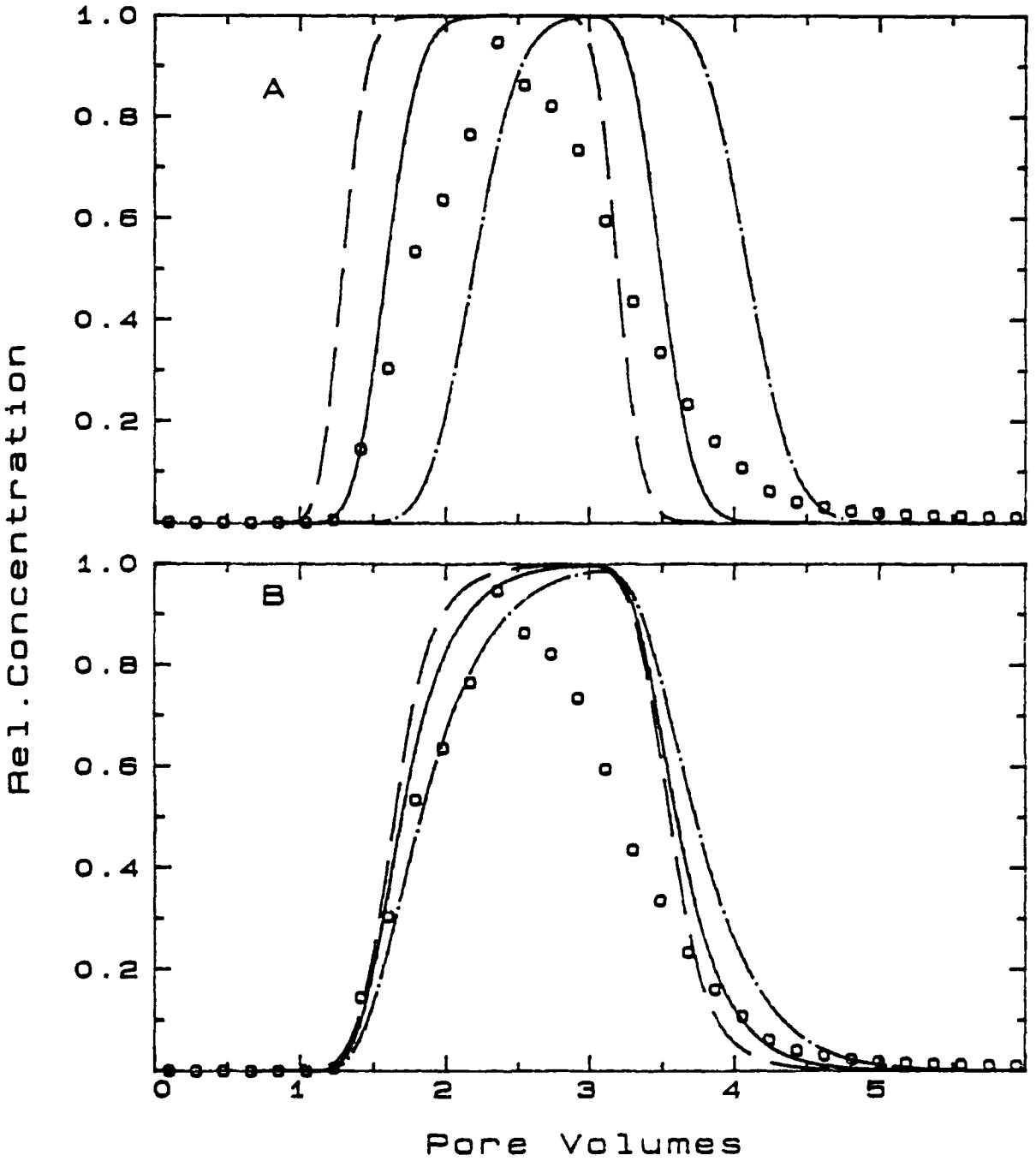


Figure 56. Simulated and Experimental Hydrazinium BTC are Presented for E2 Soil Columns with High q and Pulse Application of Solution with Medium C_o. In Section A, only Physical Sorption is Considered in the Model, and Simulated BTC's are Given for k_f/k_b Ratios of 0.3 (Broken Line), 0.6 (Smooth Line) and 1.2 (Dotted Broken Line). In Section B, Physical Non-Specific Sorption Reaction Mechanisms are Included in the Model, and Simulated BTC's are Given for k_{ff}/k_{bb} Ratios of 0.125 (Broken Line), 0.250 (Smooth Line), and 0.500 (Dotted Broken Line).

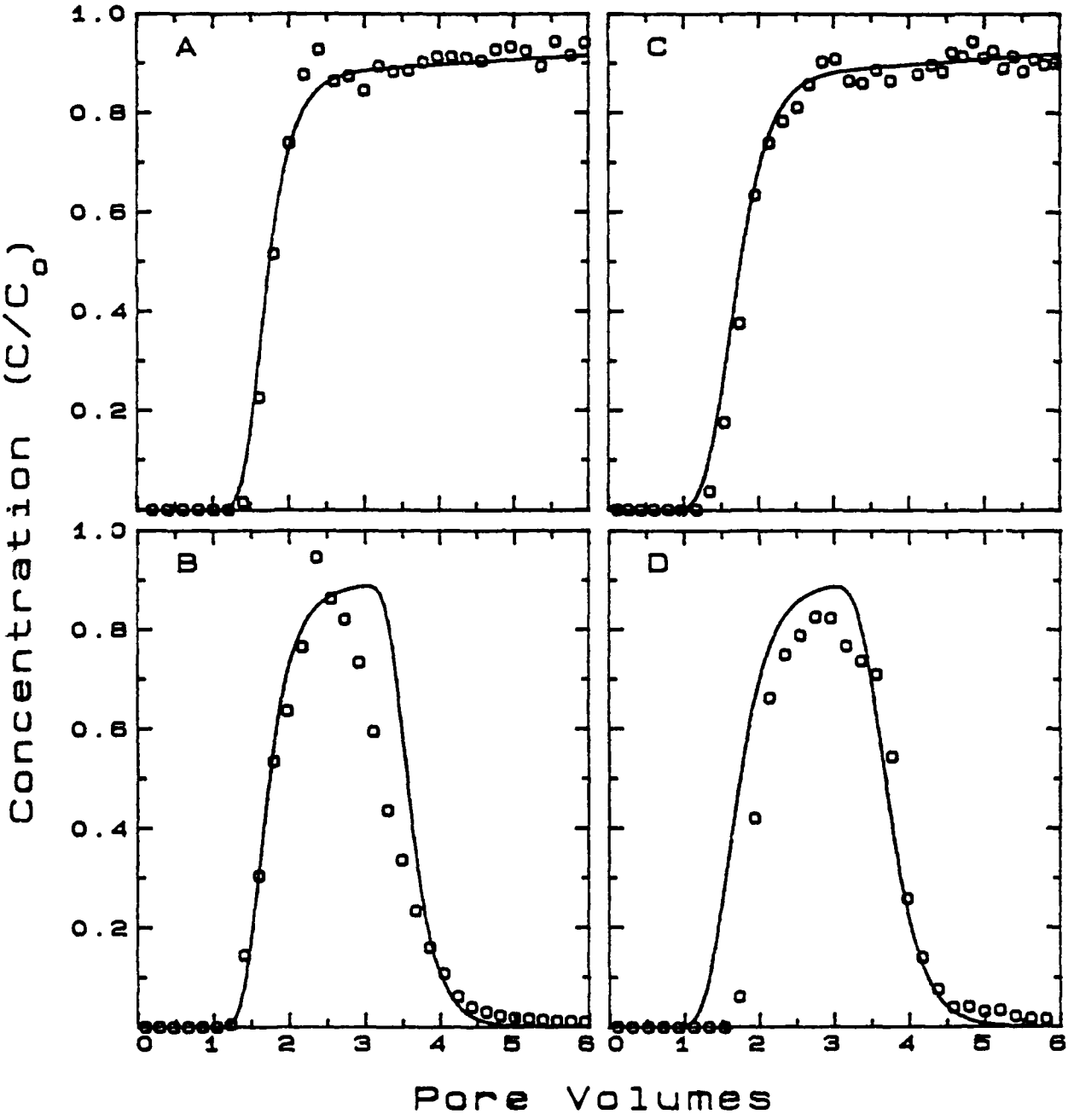


Figure 57. Experimental and Optimized Simulated Hydrazinium BTC for Columns of E2 Soil that Received Input Solution with High C_o (Using Optimized Reaction Parameters from Table 46) are Presented for Continuous Input with High q (A) and Low q (C) as Well for Pulse Input with High q (B) and Low q (D).

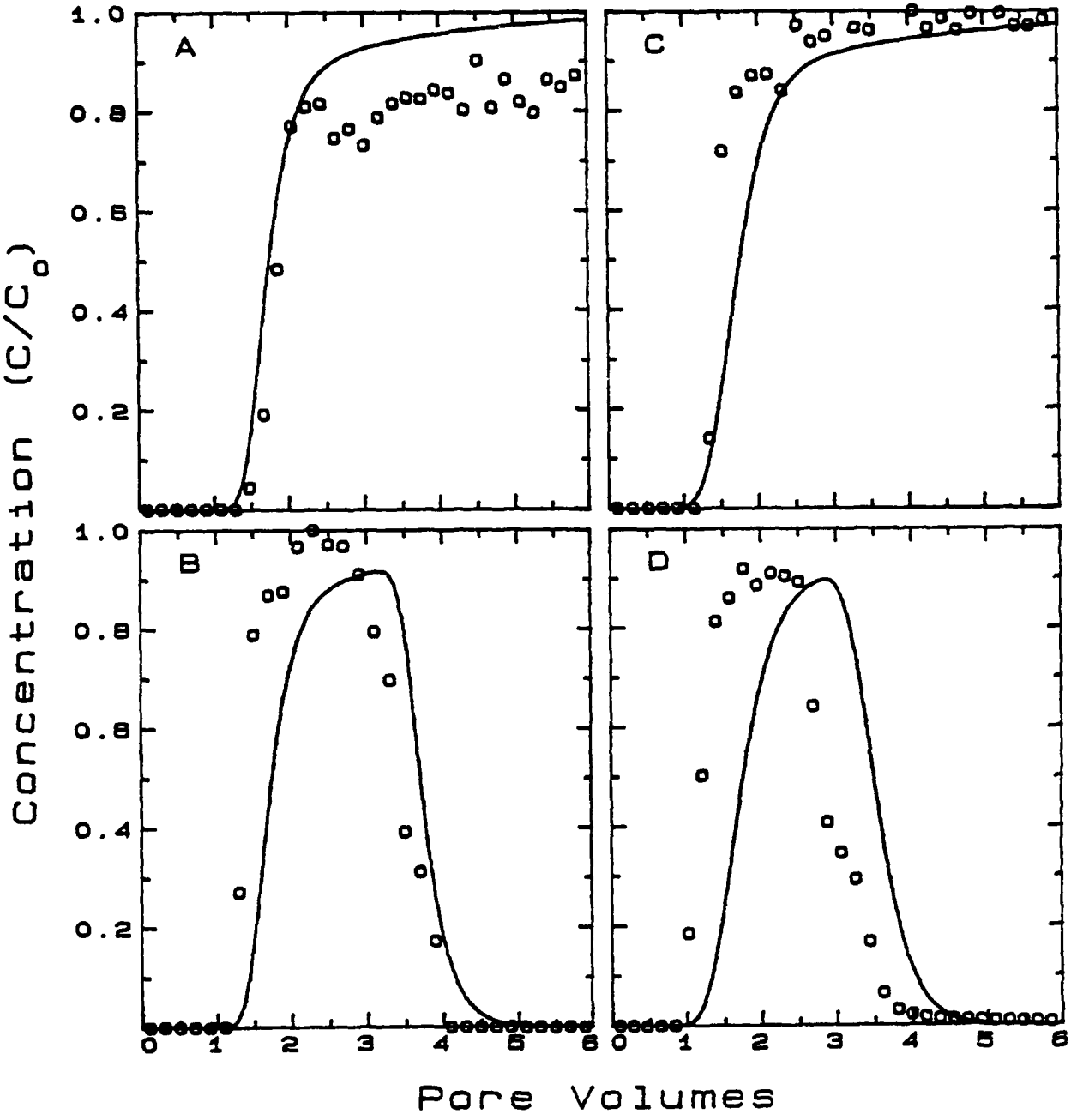


Figure 58. Experimental and Optimized (Using Reaction Parameters from Table 46) Simulated Hydrazinium BTC for Columns of E2 Soil that Received Input Solutions with Low C_o are Presented for Continuous Input with High q (A) and Low q (C) as Well as for Pulse Input with High q (B) and Low q (D).

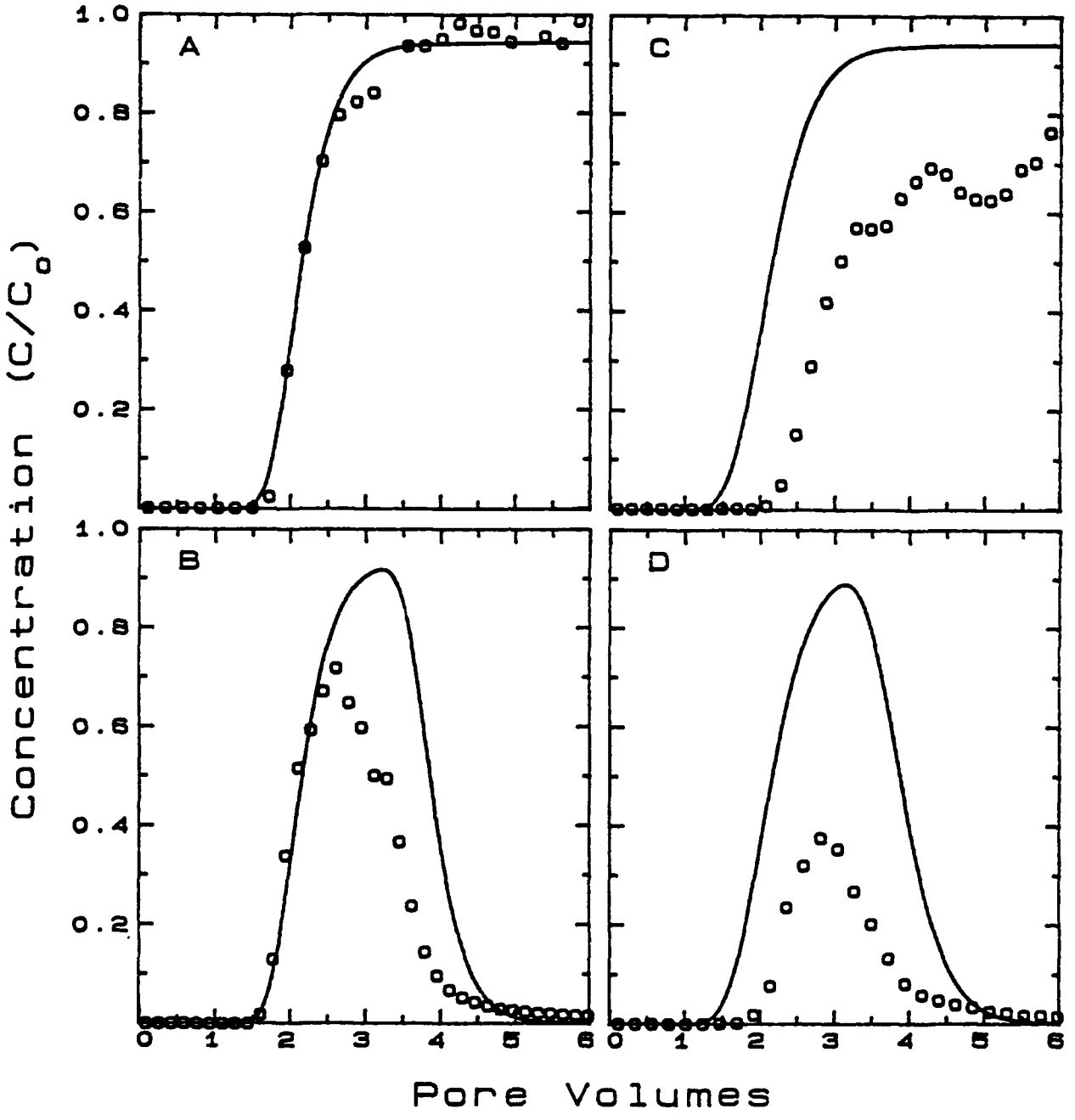


Figure 59. Experimental and optimized (used reaction parameters from Table 46) simulated hydrazinium BTC for columns of El soil that received input solutions with medium C_o are presented for continuous input with high q (A) and low q (C) as well as for pulse input with high q (B) and low q (D).

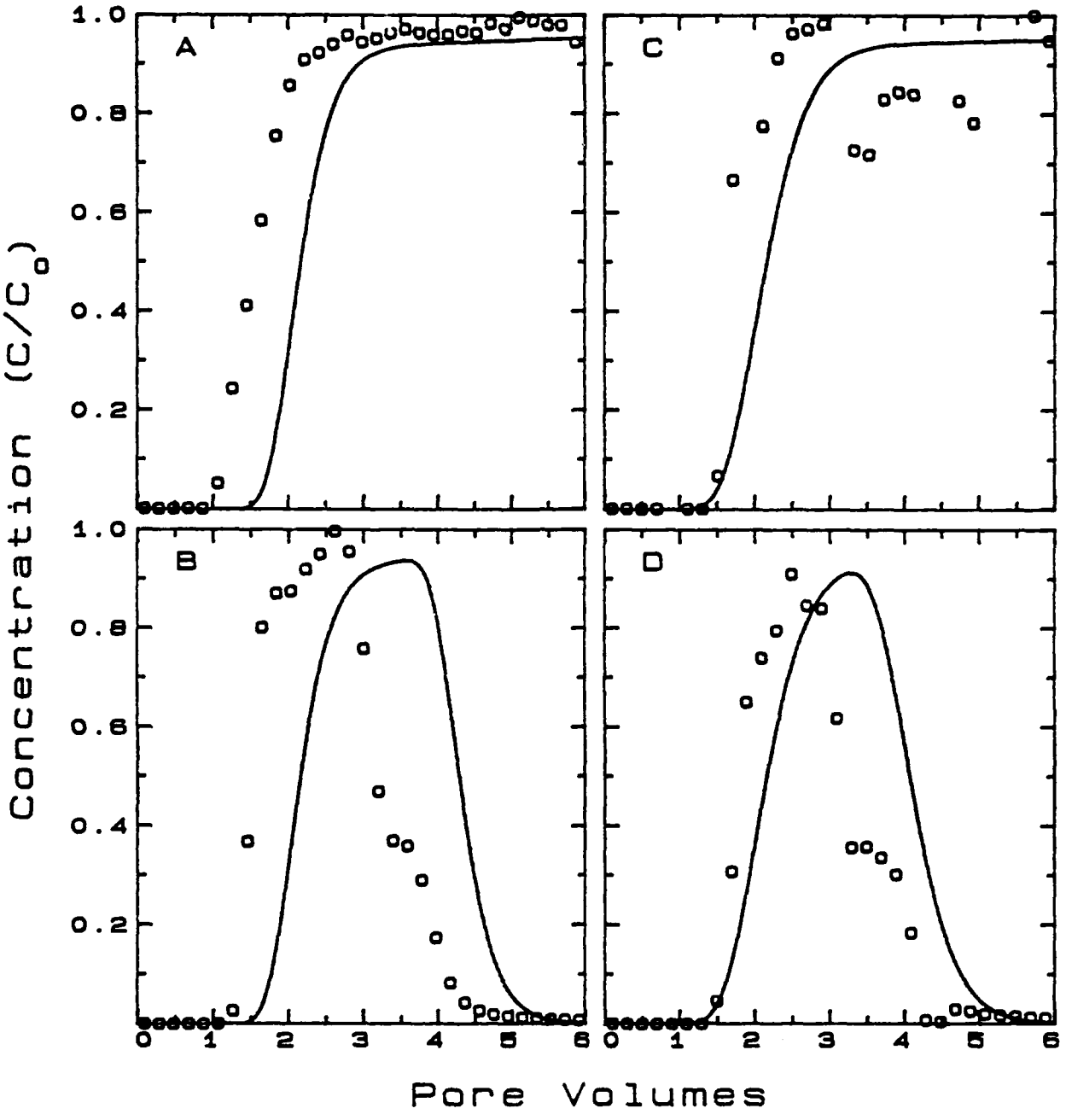


Figure 60. Experimental and Optimized (Used Reaction Parameters from Table 46) Simulated Hydrazinium BTC for Columns of El Soil that Received Input Solutions with High C_o are Presented for Continuous Input with High q (A) and Low q (C) as Well as for Pulse Input with High q (B) and Low q (D).

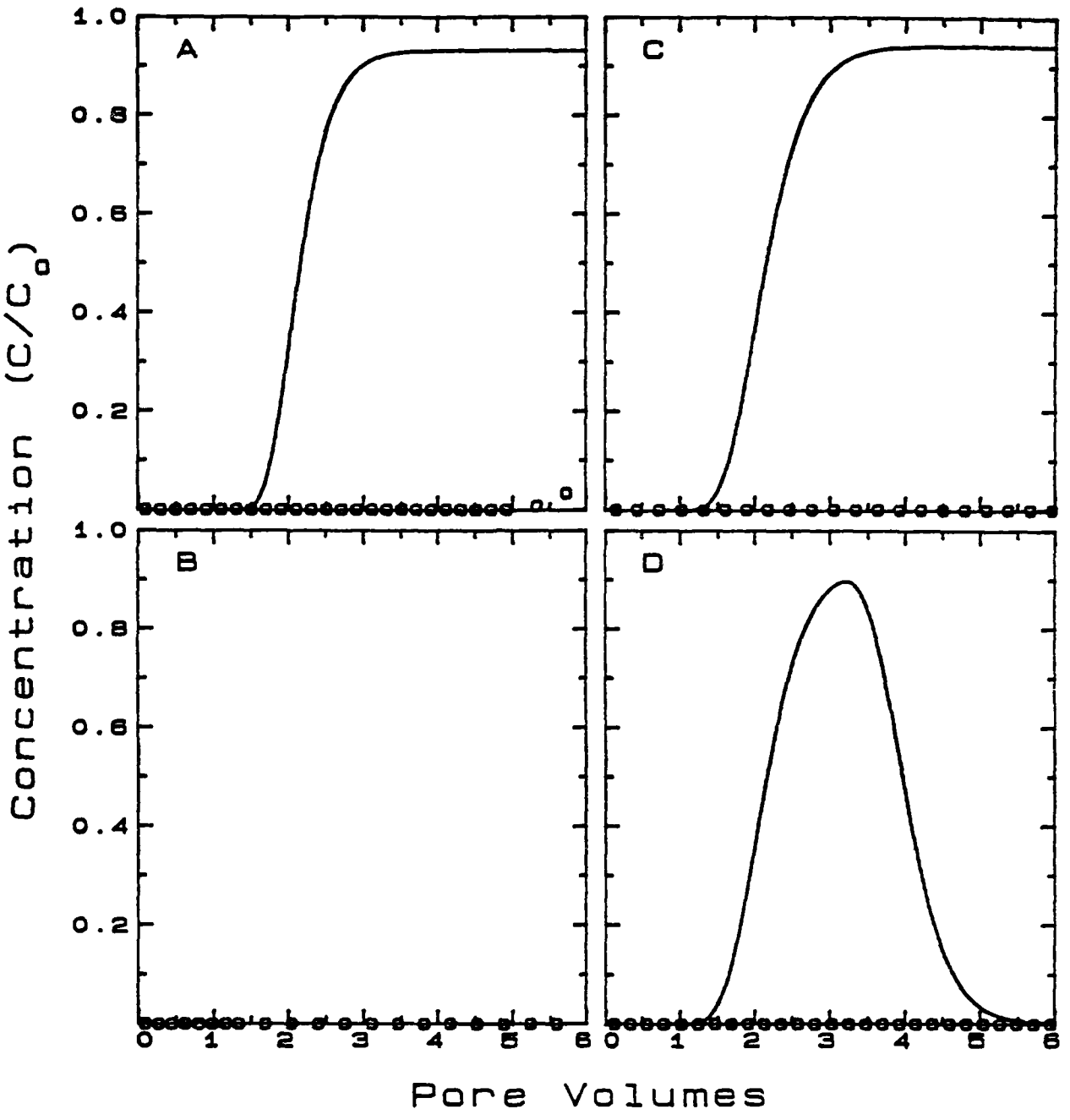


Figure 61. Experimental and Optimized (Used Reaction Parameters from Table 46) Simulated Hydrazinium BTC for Columns of El Soil that Received Input Solutions with Low C_o are Presented for Continuous Input with High q (A) and Low q (C) as Well as for Pulse Input with High q (B) and Low q (D).

indicated no recovery of $N_2H_5^+$ molecules in effluent from columns with pulse and continuous influent application and two pore water velocities. Thus, the model did not describe experimental data for the case where low C_o influent was applied to E1 soil.

4. Conclusions

A convective-dispersive chemical transport model that includes 3 kinetic reactions between hydrazinium and soil components was developed to describe the movement of hydrazinium applied in solution to columns of water-saturated soil during steady liquid flow. Hydrazinium ions in the solution phase of the soil were assumed to be subject to linear reversible sorption (i.e. cation exchange). Physically sorbed ions were assumed to be equally susceptable to reversible nonspecific sorption (tightly-bound sorption) and irreversible chemisorption. Input parameters required for the model require a hydrodynamic dispersion coefficient D, soil water content θ, Darcy liquid flow velocity q, forward and backward rate coefficients, k_f and k_b, for physical sorption, forward and backward rate coefficients, k_{ff} and k_{bb}, for nonspecific sorption, a rate coefficient k for chemisorption, and the maximum concentration S_{3m} for chemisorbed hydrazinium that the soil matrix can support. Sensitivity analysis and a calibration procedure of hydrazinium BTC were used to determine the reaction parameters for E1 and E2 subsoil materials for Arredondo fine sand. Calibration was performed by curve-fitting the model to BTC's corresponding to high pore water velocity and medium concentration of hydrazinium in applied influent for these 2 soils.

Kinetic or local nonequilibrium assumptions for the consecutive physical and nonspecific sorption processes were observed to be essential for describing tailing and general shapes of hydrazinium BTC's. Optimized values for rate coefficients for physical and non-specific sorption processes were such that backward rate coefficients were greater than forward rate coefficients, i.e. $k_b > k_f$ and $k_{bb} > k_{ff}$.

Incomplete recovery of applied hydrazinium in column effluent was predicted using the irreversible chemisorption sink term in the model when influent solution contained high concentrations of hydrazinium.

When the influent solutions contained low or medium hydrazinium concentrations, the chemisorption sink term underestimated irreversible removal of chemical from the solution phase of the soil.

The specific model as given by Equations [7] and [8] arose after many attempts to alter relationships between dissolved, physically sorbed, nonspecifically sorbed, and chemisorbed conceptual forms of $N_2H_5^+$ in the soil. The conceptual linkage of these forms of $N_2H_5^+$ in the soil was observed to be very important in describing data from soil columns.

The mathematical transport model as given by equations [7] and [8] was effective in describing the time of hydrazinium breakthrough, the relatively steep slopes for the initial portion of BTC's, maximum concentrations in the effluent, and general shapes of hydrazinium BTC's for columns of E1 and E2 subsoil materials when influent solutions contained high concentrations. In contrast, hydrazinium BTC for columns that received input solutions with low and medium concentrations exhibited small slopes for the initial portion of BTC's and fractions of applied hydrazinium not recovered in the effluent were disproportionately smaller than for BTC's corresponding to high influent concentration. Thus the model described the initial hydrazinium breakthrough in the column effluent when low and medium influent concentrations were used, but it failed to describe the shape and maximum concentrations for hydrazinium BTC's for those cases.

D. MATHEMATICAL MODEL FOR TWO-DIMENSIONAL TRANSPORT OF WATER AND SOLUTE IN SOIL

1. Description of Model

A two-dimensional numerical model was developed to describe transient solute and transient water transport in soil that receives leakage of hydrazine fuel from a subsurface storage tank and discharges groundwater through a drain located at some lateral distance L downstream from a leaking storage tank. A rectangular flow region (Figure 62) of depth $z = D$ and lateral dimension $x = L$ is considered. The soil surface occurs at the upper boundary and an impervious clay layer forms the lower

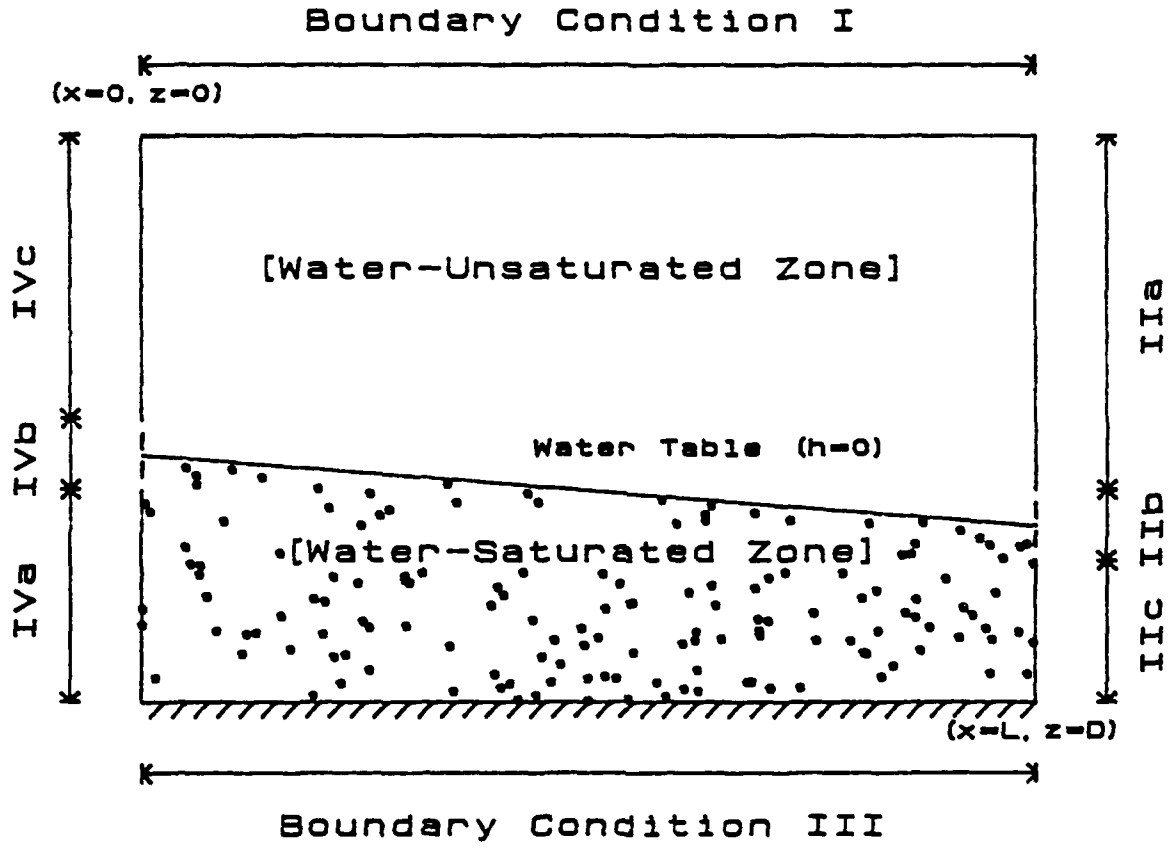


Figure 62. Schematic Diagram of Two-Dimensional Cross-Section of a Rectangular Area (with Unit Thickness) of Soil Which Receives Seepage Input of Leakage from a Storage Tank on the Upstream Side (Represented by Boundary Condition IVB) and Discharges Groundwater Through a Drain (Given by Boundary Condition IIB) on the Downstream Side. The Top Side Coincides with the Soil Ssurface and is Subject to Periodic Infiltration from Rainfall. An Impervious Clay Layer is Assumed to be Located at a Depth Z = D. The Horizontal Length of the Rectangular Flow Region is Given as L.

boundary. The upper boundary is subject to infiltration of water from periodic rainfall and/or infiltration of hydrazine fuel. Acid soil and water conditions are assumed such that hydrazine occurs predominantly in the ionic form $N_2H_5^+$. The model allows regions of water-saturation and unsaturation to occur simultaneously in the soil flow region.

Two-dimensional transient water flow in isotropic, homogeneous porous media is described by the continuity equation (Reference 48)

$$\psi \partial h/\partial t = \partial(K\ \partial h/\partial x)/\partial x + \partial(K\ \partial h/\partial z)/\partial z - \partial K/\partial z \qquad [11]$$

where h is soil water pressure head (m of water), x is horizontal distance (m), z is vertical depth (m), t is time (s), K is hydraulic conductivity (m s^{-1}) which tends to decrease as h becomes increasingly negative, and ψ is the specific soil water capacity. The parameter ψ is defined as the slope of the soil-water characteristic curve $\psi = \partial\theta/\partial h$ where θ is the volumetric water content ($m^3 m^{-3}$) of the soil. The term on the left-hand side of Equation [11] may also be expressed simply as $\partial\theta/\partial t$. Equation [11] is simply a mathematical statement of conservation of mass. Solution of the water equation requires that experimental data be provided to establish relationships for $\theta = F(h)$ and $K = G(h)$ where F(h) and G(h) are functions of water pressure head over the range $h \leq 0$. For $h \geq 0$ or conditions of water saturation, θ and K parameters are assumed to have constant values θ_o and K_o, respectively. For conditions of water-saturation where $h < 0$, specific water capacity ψ varies with h. Ordinarily $\psi = 0$ for conditions of water-unsaturation where $h \geq 0$ and Equation [11] becomes elliptic in form. The approximating assumption is made here that $\psi = 10^{-4}$ cm^{-1} for the condition $h > 0$ so that Equation [11] has a parabolic form for both saturated- and unsaturated-soil conditions.

Boundary conditions for water flow in the rectangular region are given as follows:

Designator	Condition	Location	
I	periodic water infiltration	$0 < x < L$ and $z = 0$	[12]

	at the upper soil surface with application intensity ϕ with units of $cm^3\ cm^{-2}s^{-1}$; $\phi = -K[(\partial h/\partial z - 1)]$		
IIA	no lateral flow; $\partial h/\partial x = 0$	$x = L$ and $0 < z < z_1$	[13]
IIB	$h = 0$ at a subsurface drain tube; discharge from the soil to the drain occurs when $h > 0$ in the surrounding soil	$x = L$ and $z_1 < z < z_2$	[14]
IIC	no lateral flow; $\partial h/\partial x = 0$	$x = L$ and $z_2 < z < D$	[15]
III	no vertical flow due to an impervious layer along the lower boundary; $\partial h/\partial z = 1$	$0 < x < L$ and $z = D$	[16]
IVA	no lateral flow; $\partial h/\partial x = 0$	$x = 0$ and $p_2 < z < D$	[17]
IVB	lateral seepage inflow with flux $\lambda(t)$ having units of $cm^3cm^{-2}s^{-1}$ from underground storage tank; $\lambda = -K\ \partial h/\partial x$	$x = 0$ and $p_1 < z < p_2$	[18]
IVC	no lateral flow; $\partial h/\partial x = 0$	$x = 0$ and $0 < z < p_1$	[19]

The rectangular flow region is assumed to be of unit thickness (y = 1 cm) in order to calculate volumes of soil and water. Initial conditions are required to specify the pressure head $h(x,z,t = 0)$ for all internal nodes for the flow region. The location (or Cartesian coordinates) for the water table ($h = 0$) provides the demarcation between zones of negative pressure head (region of capillary fringe and water-unsaturation) above and of positive pressure head (region of water-saturation) below.

Two-dimensional solute transport coupled with linear instantaneous sorption during transient water flow in isotropic, homogeneous porous media is described by the convective-dispersive transport equation (Reference 43).

$$\partial(\theta C)/\partial t + \theta(R-1)\partial C/\partial t = \partial[\theta D_x \partial C/\partial x - q_x C]/\partial x + \partial[\theta D_z \partial C/\partial z - q_z C]/\partial z \qquad [20]$$

where C is the solute concentration (mmol cm^{-3}) in the solution phase, D_x and D_z are horizontal and vertical components of the hydrodynamic dispersion coefficient (cm^2 h^{-1}), q_x is the horizontal component of Darcy water flux (cm h^{-1}), q_z is the vertical component of Darcy water flux (cm h^{-1}) $R = 1 + m\,[\rho/\theta]KC^{m-1}$ is the solute retardation function, where K is the solute distribution coefficient (cm^3 g^{-1}) from a Freundlich sorption isotherm $S = KC^m$ for chemical equilibrium, m is a constant and S is the solute concentration (mmol g^{-1}) in the solid phase of the soil. The horizontal and vertical components for the dispersion coefficient are assumed to be

$$D_x = D_o + D_1\, q_x/\theta \quad \text{and} \quad D_z = D_o + D_1\, q_z/\theta \qquad [21]$$

where D_o is the diffusion coefficient and D_1 is the dispersivity coefficient (reference 43).

Boundary conditions for solute transport in the rectangular flow region are specified as follows:

Designator	Condition	Location	
I	zero or imposed inflow solute flux at the upper soil surface; $J_z = q_z C$ or $J_z = 0$	$0 < x < L$ and $z = 0$	[22]
IIA	zero lateral solute flux; $J_x = 0$	$x = L$ and $0 < z < z_1$	[23]
IIB	solute elution in discharge water to a subsurface drain tube; $J_x = q_x C$ or $\partial C/\partial x = 0$	$x = L$ and $z_1 < z < z_2$	[24]
IIC	zero lateral solute flux; $J_x = 0$	$x = L$ and $z_2 < z < D$	[25]
III	zero vertical solute flux; $J_z = o$	$0 < x < L$ and $z = D$	[26]
IVA	zero lateral solute flux; $J_x = o$	$x = 0$ and $p_2 < z < D$	[27]
IV B	finite inflow flux of solute with seepage from storage tank; $\lambda C_o = \theta D_x\, \partial C/\partial x - q_x C$ where C_o	$x = 0$ and $p_1 < z < p_2$	[28]

is the hydrazinium concentration of influent

IVC zero lateral solute flux; $J_x = o \quad x = 0$ and $0 < z < p_1$ [29]

Vertical and horizontal components for solute flux are given as

$$J_z = -\theta D_z \, \partial C/\partial z + q_z C \qquad [30]$$

and

$$J_x = -\theta D_x \partial C/\partial x + q_x C. \qquad [31]$$

Initial conditions are required to specify C(x,z,t=0) for all internal nodes for the flow region.

The rectangular flow region of soil was divided into equally spaced nodes such that $\Delta x = \Delta z$. The water equation was first solved to give h(x,z,t) and then the solute transport equation was solved to give C(x,z,t). An Iterative Alternating Direction Implicit (IADI) method (Reference 61) was used to solve Equation [11] for water pressure head h(x,a,t) subject to conditions given by Equations [12] through [19]. Two finite difference approximations were used alternatively, one which is implicit in the x-direction to advance the calculations in time (n to n+1) along rows in the spatial grid of nodes from iteration level m to m+1.

$$^{m+1}\, h^{n+1}_{i-1,j} \, [- F_1]$$

$$^{m+1}\, h^{n+1}_{i,j} \, [F_1 + F_2 + F_3 + F_4 + G]$$

$$^{m+1}\, h^{n+1}_{i+1,j} \, [- F_2]$$

$$= h^{n}_{i,j-1} \, [- F_3]$$

$$+ h^{n}_{i,j} \, [- F_1 - F_2 - F_3 - F_4 + G]$$

$$+ h^{n}_{i,j+1} [F_4]$$

$$+ h^{n}_{i-1,j} [F_1]$$

$$+ h^{n}_{i+1,j} [F_2]$$

$$^{m+1}_{+} h^{n+1}_{i,j-1} [F_3]$$

$$^{m}_{+} h^{n+1}_{i,j+1} [F_4]$$

$$+ 2\Delta x (F_3 - F_4) \qquad [32]$$

where $F_1 = K^{n+\frac{1}{2}}_{i-\frac{1}{2},j}$, $F_2 = K^{n+\frac{1}{2}}_{1+\frac{1}{2},j}$, $F_3 = K^{n+\frac{1}{2}}_{i,j-\frac{1}{2}}$, $F_4 = K^{n+\frac{1}{2}}_{i,j+\frac{1}{2}}$, $G = \gamma\psi^{n+\frac{1}{2}}_{i,j}$, and $\gamma = 2(\Delta x)^2/\Delta t$ and a second which is implicit in the z-direction to advance the calculations in time (n to n+1) along columns in the spatial grid of nodes from the iteration level m+1 to m+2.

$$^{m+2} h^{n+1}_{i,j-1} [- F_3]$$

$$^{m+2} h^{n+1}_{1,j} [F_1 + F_2 + F_3 + F_4 + G]$$

$$^{m+2} h^{n+1}_{i,j+1} [- F_4]$$

$$= h^{n}_{i-1,j} [F_1]$$

$$+ h^{n}_{i,j} [- F_1 - F_2 - F_3 - F_4 + G]$$

$$+ h^{n}_{i+1,j} [F_2]$$

$$+ h^{n}_{i,j-1} [F_3]$$

$$+ h^{n}_{i,j+1} [F_4]$$

$$^{m+1} h^{n+1}_{i-1,j} [F_1]$$

$$+ \; h^{m+1}_{i+1,j}\,[F_2]^{n+1}$$

$$+ 2\Delta x\,[F_3 - F_4]. \qquad [33]$$

Subscript i refers to horizontal distance x, subscript j refers to vertical distance z, and superscript n refers to time t. Equations [32] and [33] provide a system of matrix equations which have a tridiagonal coefficient matrix and, thus can be solved using the Thomas (Reference 61) algorithm. Equation [32] was first applied to all rows of spatial nodes in the rectangular flow region and equation [33] was then applied to all columns of spatial nodes. This two-step process advanced the solution from the current time $t = n\Delta t$ to a future time $t = (n+1)\Delta t$. Iteration was then applied, and the two-step procedure was repeated sufficient times to give a convergent solution for $h(x,z,t)$ in the rectangular flow region. The iterative two-step procedure was then advanced to the next time step.

Upon completion of the solution, values obtained for $h(x,z,t)$ were used to calculate horizontal and vertical components for Darcy water flux

$$q_x = -K[\partial h/\partial x] \qquad [34]$$

and

$$q_z = -K[(\partial h/\partial z) - 1], \text{ respectively.} \qquad [35]$$

This information for q_x and q_z was then used as input to the numerical solution of equation [20] for solute transport.

After the water equation was solved, an IADI finite difference method (Reference 61) was then used to solve equation [20] for values of $C(x,z,t)$ in the rectangular flow region. Two finite difference approximations for equation [20] were used alternatively, one which is implicit in the x-direction to advance the computations in time (n to n+1) along rows in the spatial grid from iteration level m to m+1.

$$C^{m+1}_{i-1,j}\,[-A_1 - B_1]^{n+1}$$

$$
\begin{aligned}
&{}^{m+1}_{+}\, C^{n+1}_{i,j}\,[A_1 + A_2 + A_3 + A_4 + B_1 - B_2 + B_3 - B_4 + E] \\
&{}^{m+1}_{+}\, C^{n+1}_{i+1,j}\,[-A_2 + B_2] \\
&= C^{n}_{i,j-1}\,[A_3 + B_3] \\
&+ C^{n}_{i,j}\,[-A_1 - A_2 - A_3 - A_4 - B_1 + B_2 - B_3 + B_4 + E] \\
&+ C^{n}_{i,j+1}\,[A_4 + B_4] \\
&+ C^{n}_{i-1,j}\,[A_1 + B_1] \\
&+ C^{n}_{i+1,j}\,[A_2 - B_2] \\
&{}^{m}_{+}\, C^{n+1}_{i,j-1}\,[A_3 + B_3] \\
&{}^{m}_{+}\, C^{n+1}_{i,j+1}\,[A_4 - B_4]
\end{aligned}
\qquad [36]
$$

where $A_1 = [\theta D_x]^{n+\frac{1}{2}}_{i-\frac{1}{2},j}$, $A_2 = [\theta D_x]^{n+\frac{1}{2}}_{i+\frac{1}{2},j}$, $A_3 = [\theta D_z]^{n+\frac{1}{2}}_{i,j-\frac{1}{2}}$, $A_4 = [\theta D_z]^{n+\frac{1}{2}}_{i,j+\frac{1}{2}}$, $B_1 = (\Delta x/2)[q_x]^{n+\frac{1}{2}}_{i-\frac{1}{2},j}$, $B_2 = (\Delta x/2)[q_x]^{n+\frac{1}{2}}_{i+\frac{1}{2},j}$, $B_2 = (\Delta x/2)[q_x]^{n+\frac{1}{2}}_{i-\frac{1}{2},j}$, $B_3 = (\Delta x/2)[q_z]^{n+\frac{1}{2}-}_{i,j\frac{1}{2}}$, $B_4 = (\Delta x/2)[q_z]^{n+\frac{1}{2}}_{i,j+\frac{1}{2}}$, and $E = \gamma\theta^{n+\frac{1}{2}}_{i,j} R^{n+\frac{1}{2}}_{i,j}$.

and a second which is implicit in the z-direction to advance the computations in time (n to n+1) along columns in the spatial grid from iteration level m+1 to m+2.

$$
\begin{aligned}
&{}^{m+2}\, C^{n+1}_{i,j-1}\,[-A_3 - B_3] \\
&{}^{m+2}_{+}\, C^{n+1}_{i,j}\,[A_1 + A_2 + A_3 + A_4 + B_1 - B_2 + B_3 - B_4 + E] \\
&{}^{m+2}_{+}\, C^{n+1}_{i,j+1}\,[-A_4 + B_4]
\end{aligned}
$$

$$= C^{n}_{i-1,j} \left[A_1 + B_1\right]$$

$$+ C^{n}_{i,j} \left[- A_1 - A_2 - A_3 - A_4 - B_1 + B_2 - B_3 + B_4 + E\right]$$

$$+ C^{n}_{i+1,j} \left[A_2 - B_2\right]$$

$$+ C^{n}_{i,j-1} \left[A_3 + B_3\right]$$

$$+ C^{n}_{i,j+1} \left[A_4 - B_4\right]$$

$$^{m+1} C^{n+1}_{i-1,j} \left[A_1 + B_1\right]$$

$$^{m+1} C_{i+1,j} \left[A_2 - B_2\right] \qquad [37]$$

It is important to note that by definition $\theta^{n+\frac{1}{2}}_{i,j}[R^{n+\frac{1}{2}}_{i,j}-1] = \rho K$. Mass balance errors were computed for simulations of both water movement and of solute transport.

2. Results from Simulated Water and Solute Transport

The computer program for the two-dimensional model describing transient water flow and transient solute transport in soil was evaluated using a simple case of transient solute transport during steady water flow in the vertical direction. Water saturation was maintained within a rectangular flow region (Figure 62) having left-hand and right-hand boundaries with no lateral water flow. Identical water flux values of inflow and outflow were imposed at the upper and lower boundaries respectively. Breakthrough curves (BTC) for the outflow water were simulated for the case where the concentration of a conservative chemical solute in the inflow water was increased step-wise from zero to C_o. A comparison (results not presented here) between the simulated BTC and BTC obtained using a well-known analytical mathematical model showed excellent agreement indicating that the computer code was functioning properly for that simple case of solute transport.

In order to demonstrate capabilities of the 2-dimensional model, water flow and solute transport were simulated for the case of infiltration of an aqueous solution with concentration C_o = 100 mole l^{-1} with an application intensity ϕ = 2 cm h^{-1} over a 5-cm wide zone at the soil surface (Figure 62). The solute applied was assumed to be conservative (i.e. no sorption occurred) such that the retardation function R was unity. Lakeland fine sand topsoil (Reference 63) with a bulk density of 1.65 Mg m^{-3}, saturated water content of 0.352 m^3 m^{-3}, and saturated hydraulic conductivity of 3.43 x 10^{-5} m s^{-1} was selected as the soil for the rectangular flow zone. The Van Genuchten (Reference 62) method was used to provide smooth estimates of $\theta(h)$, Kh) and $\psi(h)$ functions (Figure 63) for the Lakeland soil. Parameters α = 0.02031, η = 4.190, and residual water content of 0.086 m^3 m^{-3} were used in the calculations. The depth of the zone was 50 cm and width was 25 cm. Aqueous solution was applied to the soil over the left-hand side of the upper boundary ($0 < x < 5$ cm and $z = 0$ cm). The initial water suction head in the soil was 100 cm of water and solute was absent from the initial soil solution. The node spacing used in the model was $\Delta x = \Delta z = 1$ cm. Initially time steps were $\Delta t = 0.1$ s and were gradually increased to a maximum of Δt =

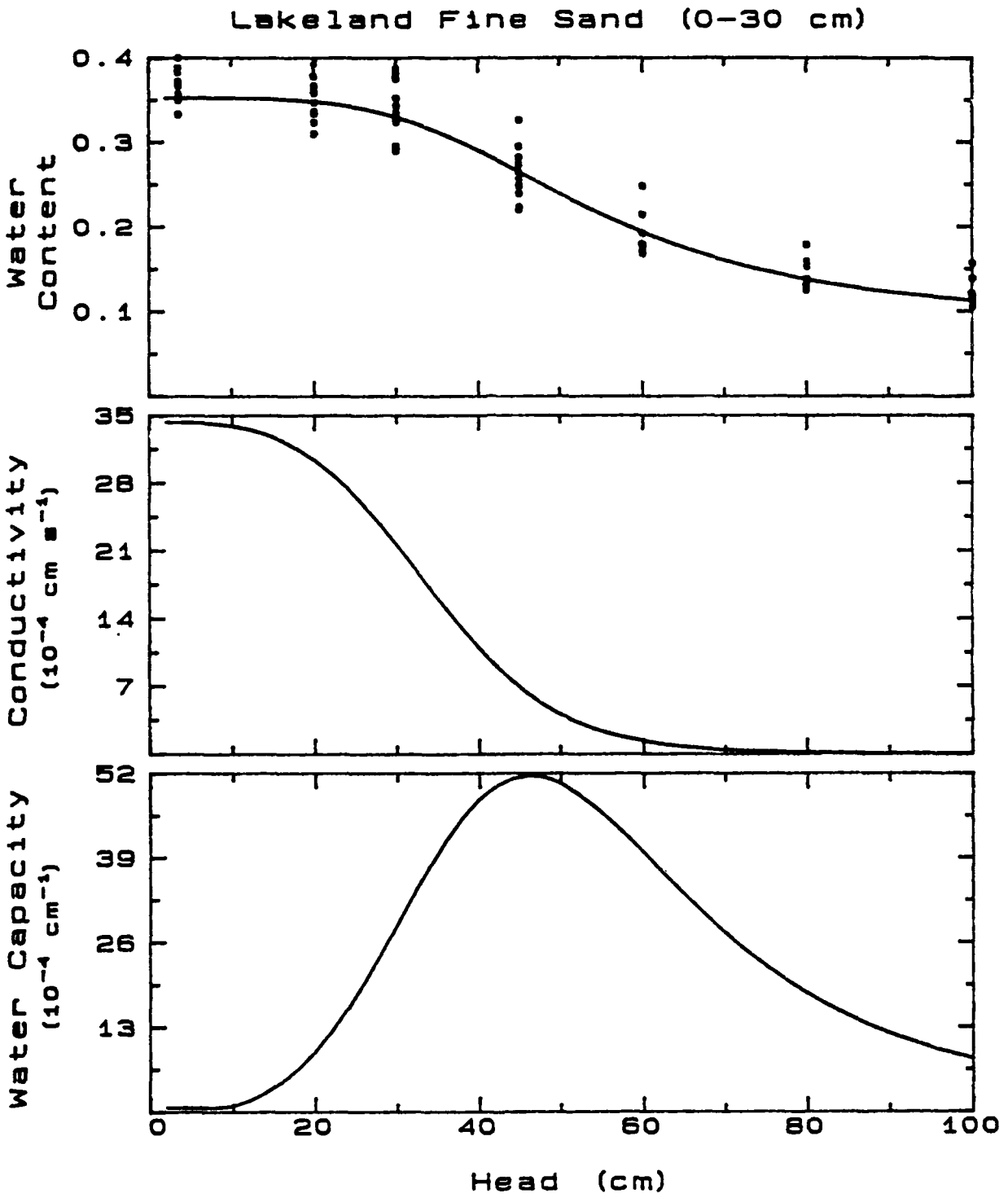


Figure 63. Graphs of Volumetric Water Content (Θ), Hydraulic Conductivity (K), and Water Capacity (ψ) Versus Water Suction Head (h) for Lakeland Fine Sand Topsoil. Smooth Curves Were Obtained Used the Van Genuchten (reference 32) Method and Discrete Points Represent Data from Dane et al (reference 63).

100 s as the simulation progressed. Values for the diffusion coefficient D_o and the dispersivity coefficient D_l were 4.3 x 10^{-8} cm^2 s^{-1} and 1.0 cm, respectively.

Two-dimensional distributions of water suction head h and solute concentration C in the Lakeland fine sand are presented for 1000 s (Figure 64), 6000 s (Figure 65), 12000 s (Figure 66), and 18000 s (Figure 67) after initiation of flux application of aqueous solution to a 5-cm wide strip of the upper soil surface. As expected, the soil immediately beneath the infiltration zone increased in water content (suction head decreased) with increasing time. The infiltrated water resulted in a wetting zone which progressed laterally and vertically downward through the soil. Solute concentrations in the soil solution also moved as a 2-dimensional front with similar shape to that of the advancing water front. The water front was observed to be influenced by gravity since as time progressed the rate of vertical advance of the front exceeded the rate of lateral advance. The solute front was observed to lag slightly behind the water front due to the presence of some initial soil water (0.115 m^3 m^{-3}) that did not contain solute. If the soil had been oven-dry (i.e. no soil water) initially, the solute and water fronts would have been expected to progress with similar rates in time.

Cumulative mass balance errors for this simulation were well within acceptable limits, being less for water flow than for solute transport. For water flow, errors were + 0.01, -0.17, -0.47, and -0.68 percent for 1000, 6000, 12000, and 18000 s, respectively. Errors for solute transport were + 0.12, -1.19, - 1.79, and -2.07 percent for 1000, 6000, 12000, and 18000s, respectively. These relatively low mass balance errors imply that the program was functioning properly. However, validation of the model using experimental data was beyond the scope of this project.

3. Conclusions

A finite difference technique was used to develop a two-dimensional solute transport model for conditions of transient water flow in soil. Tentative evaluation of the model indicates that the program provides expected results for the simple case of a conservative solute (such as

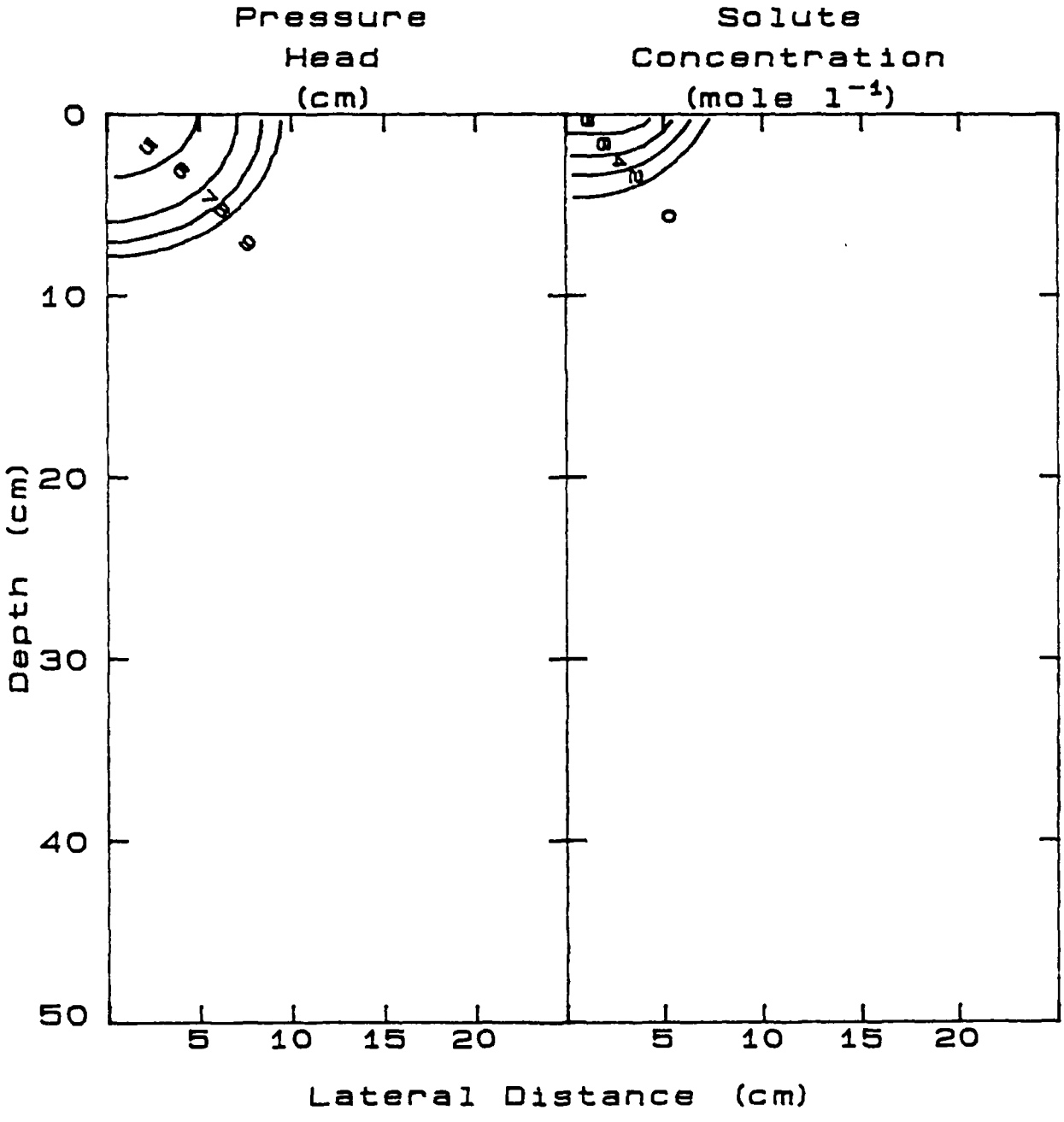


Figure 64. Simulated Distributions of Soil Water Suction Head (h) and Concentration (C) of a Conservative Solute After 1000s Infiltration of a Salt Solution at the Upper Left-Hand Soil Surface. Numbers 9, 8, 7, 6, and 5 for impressive heads indicate 100 to 90, 90 to 80, 80 to 70, 70 to 60, and 60 to 50 cm of water, respectively. Numbers 0, 2, 4, 6, and 8 for iso-concentrations designate 0 to 20, 20 to 40, 40 to 60, 60, 80, and 80 to 100 mole l^{-1}, respectively.

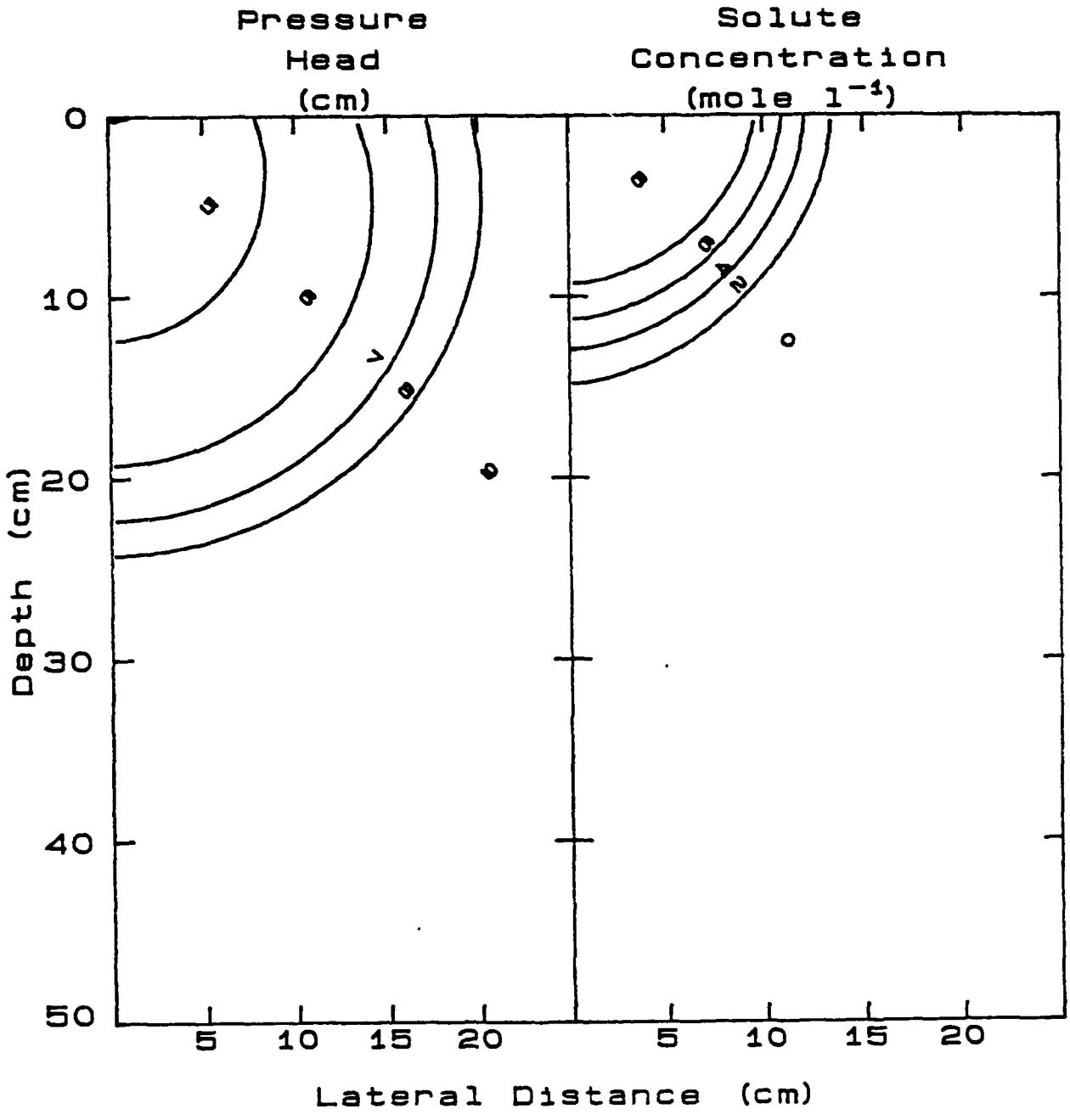


Figure 65. Simulated Distributions of Suction Head and Solute Concentration After 6000s Infiltration of a Salt Solution.

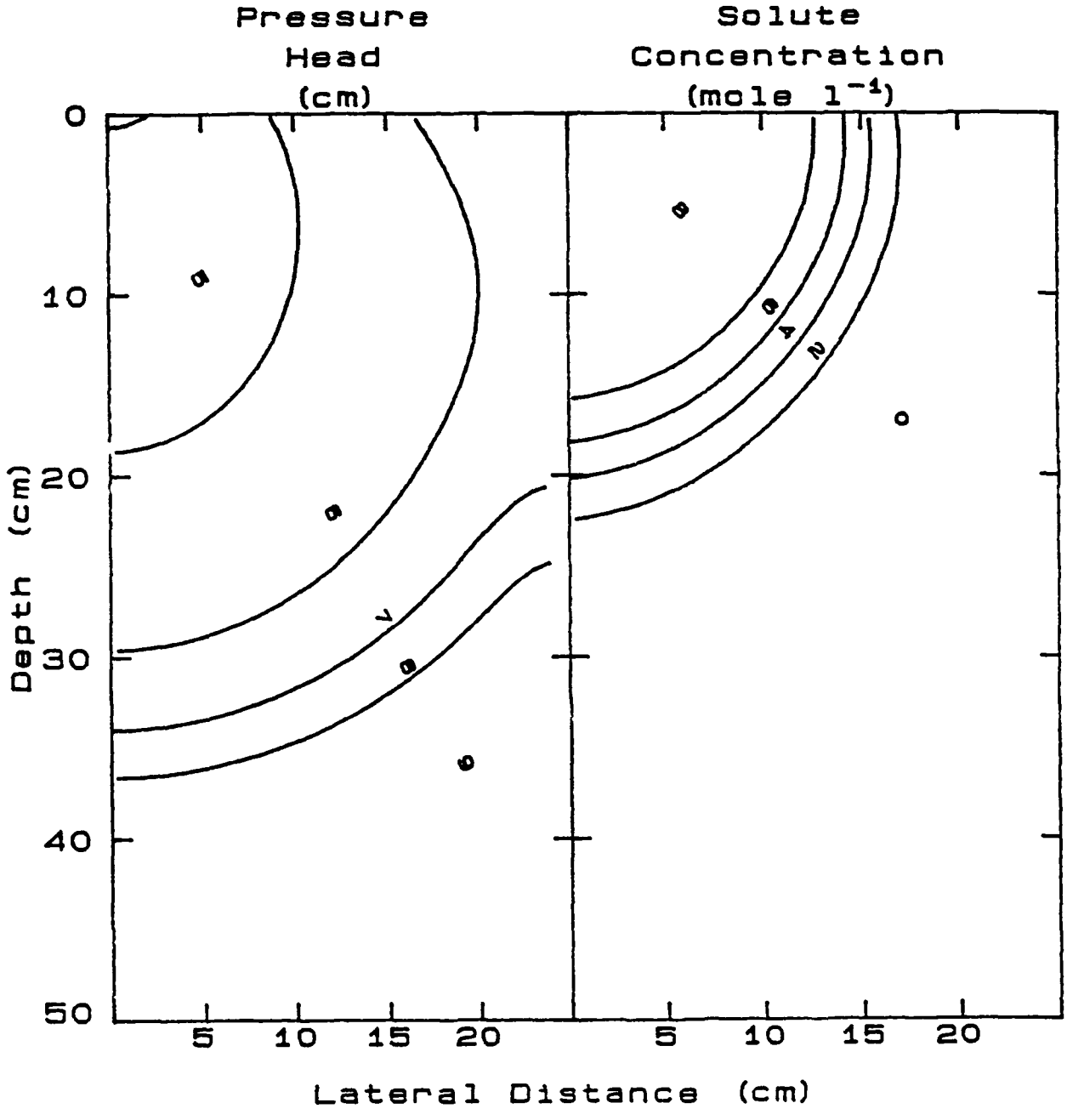


Figure 66. Simulated Distributions of Suction Head and Solute Concentration After 12000s Infiltration of a Salt Solution.

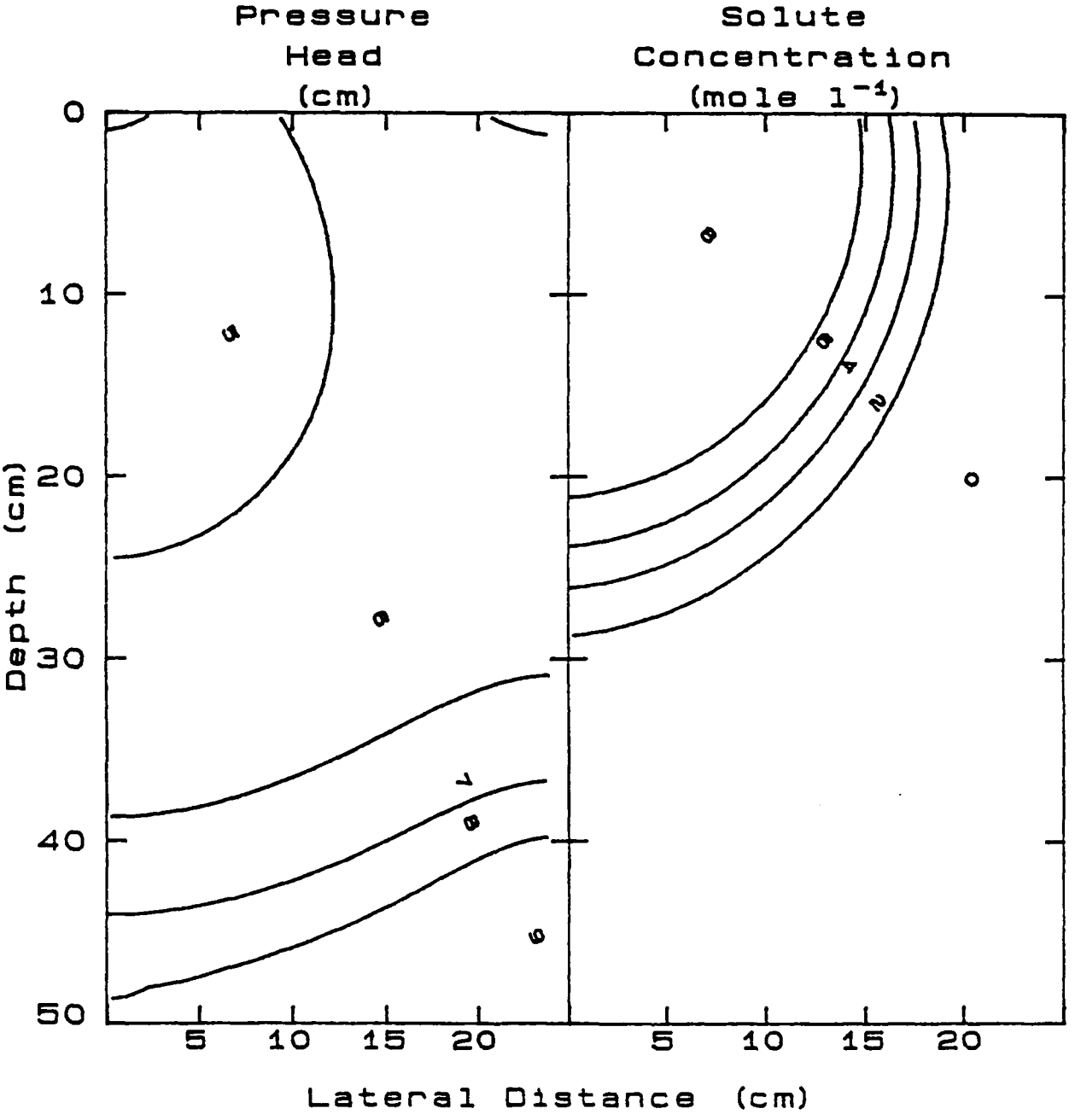


Figure 67. Simulated Distributions of Suction Head and Solute Concentration After 18000s Infiltration of a Salt Solution.

tritiated water tracer). For conditions of chemical equilibrium, the model can be used to simulate the transport of a nonconservative solute (such as hydrazinium) by assuming that a Freundlich isotherm can be used to describe the net influence of all chemical reactions that control solute concentration in the soil solution. Although in the present form of the model reaction kinetics have been omitted, the program could be modified to accommodate non instantaneous chemical reactions.

E. RECOMMENDATIONS FOR FUTURE RESEARCH

Ongoing research by doctoral graduate student Wayne Downs is underway to provide independent determination of values of rate coefficients and other pertinent reaction parameters that control the kinetic transfer of hydrazinium ions between solution and solids phases of soil from Ap, E1 and E2 horizons of Arredondo fine sand. Further understanding of the role of cation exchange upon the sorption-desorption of hydrazinium in soil is needed. Investigations using stirred slurries of soil in aqueous solution as well as flow through soil columns are needed to determine overall sorption kinetics. This information is needed to provide further evaluation of the mathematical model as well as improved understanding of the mobility of hydrazinium applied to this soil.

Column experiments in this project involved water-saturation of the soil in order to simplify conditions that might occur under a field site. This simplification was needed in order to facilitate the investigation of the importance of chemical and physical reactions upon hydrazinium transport in soil. Water-saturation of the soil profile generally assumes a shallow water table and poor lateral drainage to a stream or lake. In terms of transport of a reactive chemical, water-saturation of the soil profile represents a worst-case scenario with respect to potential groundwater contamination. A second stage of research is needed to investigate the fate and transport of $N_2H_5^+$ and N_2H_4 applied to water-unsaturated soil under conditions of intermittent rainfall. Such conditions are common in soil profiles with relatively deep water tables. Transport of chemicals through unsaturated soil generally is a slower process with longer residence times and more effective contact with

reactive soil components than flow through saturated soil. Generally speaking reactive chemicals such as pesticides, orthophosphate, and organic wastes tend to interact more completely with soil components and thus be less mobile under conditions of aerobic, water-saturation of the soil. Laboratory soil columns, large lysimeters, and small field sites should be used to evaluate the fate and transport of hydrazine during transient water flow through water-unsaturated soil. Field site data is particularly needed to provide fate and transport information in a natural setting. Two-dimensional data from carefully-selected field experiments are needed to evaluate the unvalidated 2-D transport model reported in this report. More detailed submodels for microbial degradation and chemical-physical reactions should be evaluated first under non flow conditions in batch reactors of stirred soil suspensions under water saturation conditions. Selected submodels should then be coupled to a one-dimensional convective-dispersive transport model in order to better describe water and hydrazine movement in both water-saturated and water-unsaturated soil.

Further research is needed to provide a data base of hydrazine mobility under flow conditions in columns for a wide range of soils with varying proportions of reactive components such as sesquioxides, clay minerals, and organic matter. The dependence of hydrazinium mobility and reactions in soil upon solution pH, temperature, ionic composition of the soil solution and soil aeration should be determined for a wide range of soils and water flux conditions in soil columns.

REFERENCES

1. Hayes, M.H.B., Isaacson, P.J., Chia, K.Y., Lees, A.M., and Yormah, T.B.R., Interactions of Hydrazine and of Hydrazine Derivatives with Soil Constitutents and with Soils, AFOSR-80-0032, Final Report, Air Force Office of Scientific Research, London, 23 August 1984.

2. Penneman, R.A. and Audrieth, L.F., "Quantitative Determination of Hydrazine," Analytical Chemistry, Vol. 20, No. 11, November 1948, pp. 1058-1061.

3. Ou, L.T. and Street, J.J., "Hydrazine Degradation and its Effect on Microbial Activity in Soil," Bulletin of Environmental Contamination and Toxicology, 1987, Vol. 38, pp. 179-183.

4. Slonim, A.R. and Gisclard, J.B., "Hydrazine Degradation in Aquatic Systems," Bulletin of Environmental Contamination and Toxicology, 1976, Vol 16, No. 3, pp. 301-309.

5. Gilbert, E.C. "Studies on Hydrazine. The Auto-Oxidation," J.Am. Chem. Soc. 51, 2744-2751 (1929).

6. Audrieth, L.F. and Mohr, P.H., "Autoxidation of Hydrazine: Effect of Dissolved Metals and Deactivators," Ind. and Eng. Chemistry, Vol. 43, No. 8, August 1951, pp. 1774-1779.

7. Gaunt, H. and Wetton, E.A.M., "The Reaction Between Hydrazine and Oxygen in Water," Journal of Applied Chemistry, 1966, Vol. 16, pp. 171-176.

8. Johnston, C.T. and Sposito, G., 1987, "Disorder and Early Sorrow: Progress in the Chemical Speciation of Soil Surfaces" in Future Developments in Soil Science Research (Ed. Boersma, L.L.) Publ. by Soil Sci. Soc. Am., pp 89-99.

9. Farmer, V.C., 1974, Chapter 15 in "The Infrared Spectra of Minerals" Ed. Farmer, V.C. Publ. by the Mineral Society of London.

10. Johnston, C.T., Sposito, G., and Birge, R.E., 1985, "Raman Spectroscopic Study of Kaolinite in Aqueous Suspension." Clays and Clay Minerals, 33, 483.

11. van Olphen, H. and Fripiat, J.J., 1979, Data Handbook for Clay Materials and Other Non-metallic Minerals. Pergamon Press, Oxford, 346 pp.

12. Sposito, G., Holtzclaw, K.H., Johnston, C.T., and LeVesque-Madore, C.S., 1981, "Thermodynamics of Sodium-Copper Exchange on Wyoming Bentonite at 298 K." Soil Sci. Soc. Am. J. 45:1079-1084.

13. Forman, M.L., Steele, W.H., Vanasse, G.A., 1966a, "Correction of Asymmectric Interferograms Obtained in Fourier Spectroscopy." J. Opt. Soc. Am., 56, 59-63.

14. Forman, M.L., 1966, "Fast Fourier-transform Technique and its Application to Fourier Spectroscopy." J. Opt. Soc. Am., 56, 978.

15. Forman, M.L., 1977, "Spectral Interpolation: Zero Fill or Convolution." Applied Optics. 16, 2801.

16. Prost, R., 1973, "The Influence of the Christiansen Effect on IR Spectra of Powders." Clays and Clay Min., 21, 363-368.

17. Griffiths, P.R. and de Haseth, J.A., 1986, Chapter 9 in "Fourier transform Infrared Spectroscopy," John Wiley, New York.

18. Ledoux, R.L. and White, J.L., 1964, "Infrared Study of Selective Deuteration of Kaolinite and Halloysite at Room Temperature." Science 145, 47-49.

19. Wada, K., 1967, "A Study of Hydroxyl Groups in Kaolin Minerals Utilizing Selective Deuteration and Infrared Spectroscopy." Clay Minerals, 7, 51.

20. Farmer, V.C., 1964, Science, 145, 1189.

21. Farmer, V.C. and Russell, J.D., 1964, "The Infrared Spectra of Layer Silicates." Spectrochim. Acta, 20, 1149.

22. Stubican, V. and Roy, R., 1961, "A New Approach to Assignment of Infra-red Absorption Bands in Layer-structure Silicates." Zeitschriftfur Kristallographie, 115, 200-214.

23. Wiewora, A., Wieckowski, T., Sokolowska, A., 1979, "The Raman Spectra of Kaolinite Sub-group Minerals and of Pyrophyllite." Archium Mineralogiczne, T. XXXV, 5-14.

24. Prost, R., 1984, "Etude par Spectoscopie Infrarouge a Basse Termpature des Groupes OH de Strucure de la Kaolinite, de la Dickite et de la Nacrite." Agronomic, 4, 403-406.

25. Fripiat, J.J. and Toussaint, F., 1963, "Dehyroxylation of Kaolinite. II. Conductometric Measurements and Infrared Spectroscopy." J. Phys. Chem., 67, 30-36.

26. Ledoux, R.L. and White, J.L., 1966, "Infrared Studies of Hydrogen Bonding Interaction Between Kaolinite Surfaces and Intercalated Potassium Acetate, Hydrazine, Formamide, and Urea." J. Coll. Interface Sci., 21, 127-152.

27. Kane, D.A. and Williamson, K.J., "Bacterial Toxicity and Metabolism of Hydrazine Fules," Archive of Environmental Contamination and Toxicology, 1983, Vol 12, pp. 447-453.

28. London, S.A. and Mantel, C.R., "Effect of Hydrazine on Substrate Utilization by a Strain of Enterobacter cloacae," Bulletin of Environmental Contamination and Toxicology, 1983, Vol. 30, pp. 497-504.

29. London, S.A., Mantel, C.R., Robinson, J.D., and Luking, S., "Effects of Selected Hydrazines on the Early Death Rates of Enterobacter cloacae," Bulletin of Environmental Contaminations and Toxicology, 1983, Vol. 31, pp. 360-368.

30. Mantel C. and London, S., "Adaptation of a Soil Bacterium to Hydrazine Propellants," Bulletin of Environmental Contamination and Toxicology, 1980, Vol. 25, pp. 672-770.

31. Stiefel, E.D., Newton, W.E., Watt, G.D., Hadfield, K.L., and Bulen, W.A., "Molybdoenzyme: The Role of Electrons, Protons, and Dihydrogen," Advanced Chemistry Series, 1979, Vol 162, pp. 353-388.

32. Watt, G.W. and Chrisp, J.D., "A Spectrophotometric Method for the Determination of Hydrazine," Analytical Chemistry, 1952, Vol. 24, pp. 2006-2008.

33. Reynolds, B.A. and Thomas, A.A., "A Colorimetric Method for the Determination of Hydrazine and Monomethyl Hydrazine in Blood," American Industrial Hygiene Association Journal, 1965, Vol. 26, pp. 527-531.

34. Ou, L.T., Gancarz, D.H., Wheeler, W.B., Rao, P.S.C., and Davidson, J.M., "Influence of Soil Temperature and Soil Moisture on Degradation and Metabolism of Carbofuran in Soil," Journal of Environmental Quality, 1982, Vol. 11, pp. 293-298.

35. Ou, L.T., Davidson, J.M., and Rothwell, D.F., "Response of Soil Microflora to High 2,4-D Applications," Soil Biology & Biochemistry, 1978, Vol 10, pp. 443-445.

36. Ou, L.T., Rothwell, D.F., Webster, W.B., and Davidson, J.M., "The Effect of High 2,4-D Concentrations on Degradation and Carbon Dioxide Evolution in Soils," Journal of Environmental Quality, 1978, Vol. 7, pp. 241-246.

37. Bradford, M.M., "A Rapid and Sensitive Method for the Quantitation of Microgram Quantities of Protein Utilizing the Principle of Protein-dye Binding," Analytical Biochemistry, 1976, Vol. 72, pp. 248-25.

38. Barnes, H. and Folkard, A.R., "The Determination of Nitrites," Analyst, 1951, Vol. 76, pp. 599-603.

39. Ou, L.T., "Microbial Degradation of Hydrazine," Bulletin of Environmental Contamination and Toxicology, 1987, Vol. 39, pp. 78-85.

40. Schmidt, E. W., Hydrazine and Its Derivatives. John Wiley & Sons, New York, 1984.

41. Alexander, M., Introduction to Soil Microbiology, 2nd ed., pp. 269-270, John Wiley & Sons, New York, 1977. Yashida, T. and M. Alexander, "Hydroxyamine Formation by Nitrosomonas europea," Canadian Journal of Microbiology, 1964, Vol. 10, pp. 923-926.

42. Yashida, T. and Alexander, M., "Hycroxyamine Formation by Nitrosomonas europea," Canadian Journal of Microbiology, 1964, Vol. 10, pp. 923-926.

43. Bear, Jacob. 1979. "Hydraulics of Groundwater." McGraw Hill Inc., New York, N.Y. 567 pages.

44. Braun, B.A. and Zirroli, J.A. 1983. "Environmental Fate of Hydrazine Fuels in Aqueous and Soil Environments." ESL-TR-82-45 to the U.S. Air Force Engineering and Services Center, Environics Division.

45. Downs, W.C., Mansell, R.S., Street, J.J., Bloom, S.A., and Augustijn, D.C.M. 1987. "Hydrazine Transport in Columns of Sandy Soils." Proceedings of International Conference on Impact of Physico-Chemistry on the Study, Design, and Optimization of Processes in Natural Porous Media Convened June 10-12, 1987 in Nancy, France. Pages 285-291.

46. Gee, G.W., and Bauder, J.W. 1986. "Particle-Size Analysis." p 383-411. In Arnold Klute (ed.) Methods of Soil Analysis Part 1, Second Edition. Agronomy 9 (Part 1):383-411.

47. Isaacson, P.J. and Hayes, M.H.B. 1984. "The Interaction of Hydrazine Hydrate with Humic Acid Preparations at pH 4." J. Soil Sci. 35:79-92.

48. Kirkham, Don and Powers, W.L. 1972. "Advanced Soil Physics." Wiley-Interscience, New York, see pages 415-419.

49. Leasure, C.S., and Miller, E.L. 1987. "Measurement of Hydrazine Contamination in Soils." Proceedings of the Third Conference on the Environmental Chemistry of Hydrazine Fuels Convened Sept. 15-17, 1987 in Panama City, Florida by the U.S. Air Force Engineering and Services Center, Environics Division.

50. Mansell, R.S., Selim, H.M., Kanchanasut, P., Davidson, J.M., and Fiskell, J.G.A. 1977. "Experimental and Simulated Transport of Phosphorus Through Sandy Soils." Water Resour. Res. 13:189-194.

51. Mansell, R.S. and Selim, H.M. 1981. "Mathematical Models for Predicting Reactions and Transport of Phosphorus Applied to Soil." Chapter 21 in "Modeling Wastewater Renovation," John Wiley, I.K. Iskandar (ed.). Pages 600-646.

52. Mansell, R.S., Bloom, S.A., and Downs, W.C. 1987. "Numerical Simulation of Hydrazine Transport in a Sandy Soil." Proceedings of the Third Conference on the Environmental Chemistry of Hydrazine Fuels

Convened 15-17 Sept. 1987 in Panama City, Florida by the U.S. Air Force Engineering and Services Center, Environics Division.

53. Mansell, R.S., Bloom, S.A., Selim, H.M., and Rhue, R.D. 1988. "Using Variable Exchange Selectivity Coefficients to Simulate the Transport of Multiple Cations in Soil." SSSAJ (Accepted for Publication).

54. Nelson, D.W. and Sommers, L.E. 1986. "Total Carbon, Organic Carbon, and Organic Matter." p. 539-580. In Arnold Klute (ed.) Methods of Soil Analysis Part 2, Second Edition. Agronomy 9 (Part 2):539-580.

55. Ou, L.T. and Street, J.J. 1987. "Microbial Enhancement of Hydrazine Degradation in Soil and Water." Bull Environ. Contam. Toxicol. 39:541-548.

56. Rhue, R.D. and Mansell, R.S. 1988. "The Effect of Ph on Na-Ca and K-Ca Exchange Selectivity for Cecil Sandy Loam Soil." SSSAJ (Accepted for Publication).

57. Schiessl, H.W. 1980. "Hydrazine-Rocket Fuel to Synthetic Tool." Aldrichimica Acta 13:33-40.

58. Schmidt, E.W. 1987. "One Hundred Years of Hydrazine Chemistry." Proceedings of the Third Conference on the Environmental Chemistry of Hydrazine Fuels Convened Sept. 15-17, 1987 in Panama City, Florida by the U.S. Air Force Engineering and Services Center, Environics Division.

59. Selim, H.M., Davidson, J.M., and Mansell, R.S. 1976. "Evaluation of a Two-Site Adsorption-Desorption Model for Describing Solute Transport in Soils." Proceedings of Summer Computer Simulation Conference convened July 12-14, 1976 in Washington, D.C. pages 444-448.

60. Thomas, B.P., Cummings, E., and Wiltstruck, W.H. 1985. "Soil Survey of Alachua County, Florida." U.S.D.A. Soil Conservation Service and University of Florida IFAS. Agricultural Experiment Stations, Soil Science Department, Gainesville, Florida, Pages 219-221.

61. Wang, H.F. and Anderson, M.P. 1982. "Introduction to Groundwater Modeling." W.H. Freeman and Company, San Francisco, California. 237 pages.

62. Van Genuchten, M. Th. 1980. "A Closed-Form Equation for Predicting the Hydraulic Properties of Unsaturated Soils." Soil Sci. Soc. Amer. J. 44:892-898.

63. Dane, J.H., Cassel, D.K., Davidson, J.M., Pollant, W.L., and Quisenberry, V.L. 1983. "Physical Characteristics of Soils of the

Southern Region -- Troup and Lakeland Series." Southern Cooperative Series Bulletin 262, Alabama Agricultural Experiment Station, Auburn University, Alabama. 258 pages.

APPENDIX A

SELECTED FIGURES FOR COLUMN EXPERIMENTS

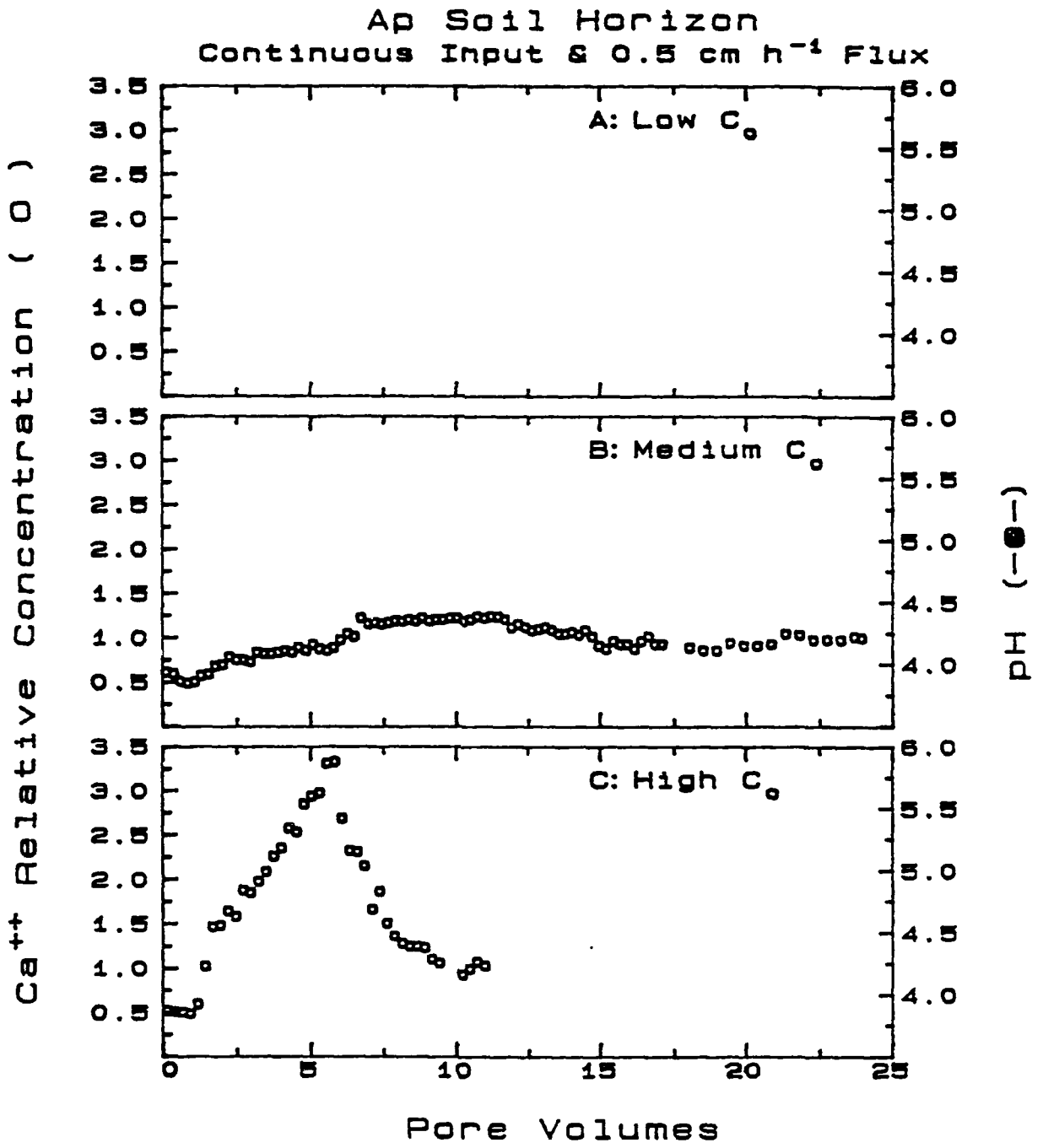


Figure 68. Effluent pH and Ca^{2+} Concentration from Columns of Ap Soil that Received Continuous Application of Influent with A. Low, B. Medium, and C. High Concentrations C_o of Hydrazinium Hydrate. Liquid Flux was $q_o = 0.5$ cm h^{-1}.

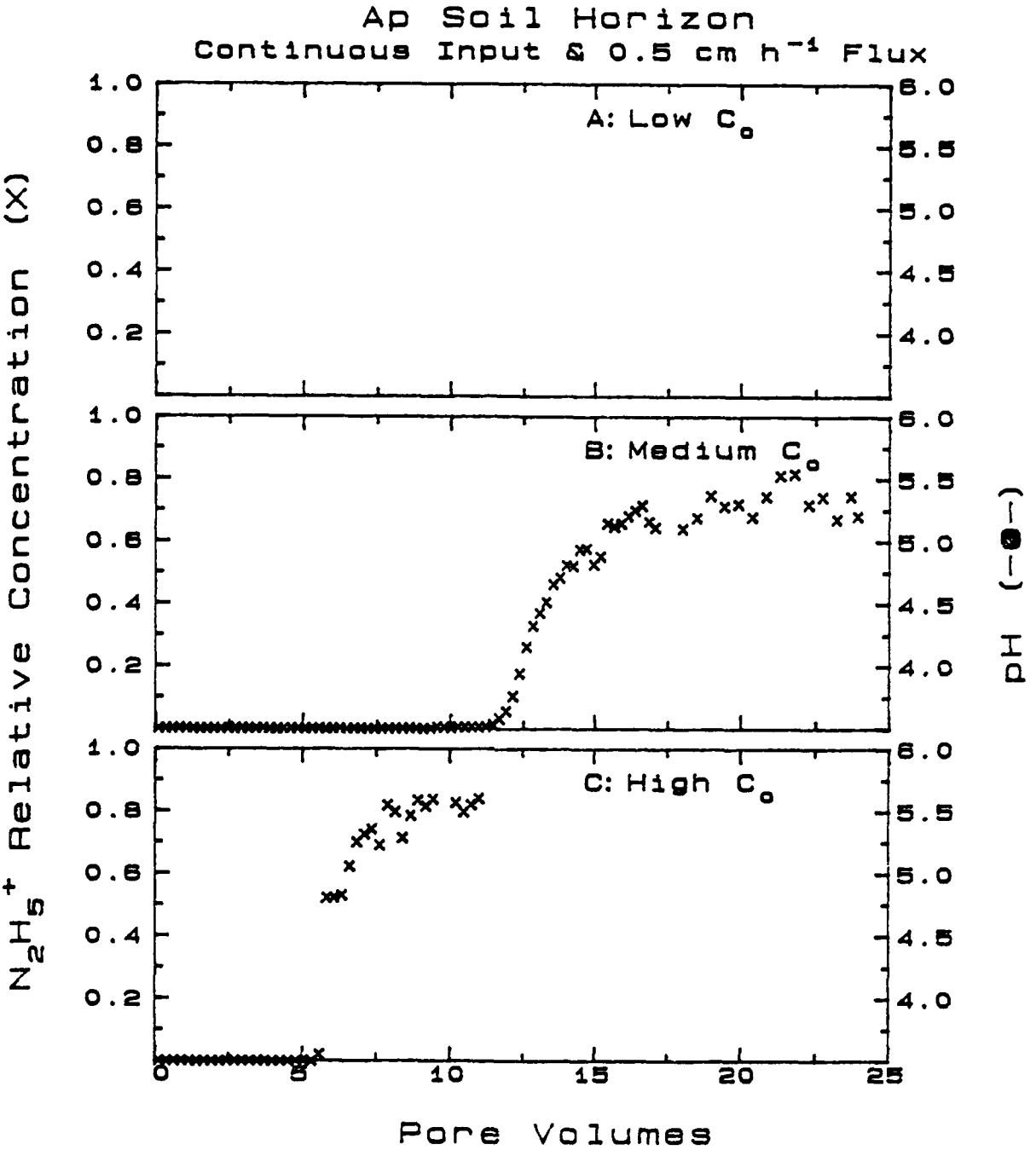


Figure 69. Effluent pH and Hydrazinium Concentration from Columns of Ap Soil that Received Continuous Application of Influent with A. Low, B. Medium, and C. High C_o. Liquid Flux was 0.5 cm h^{-1}.

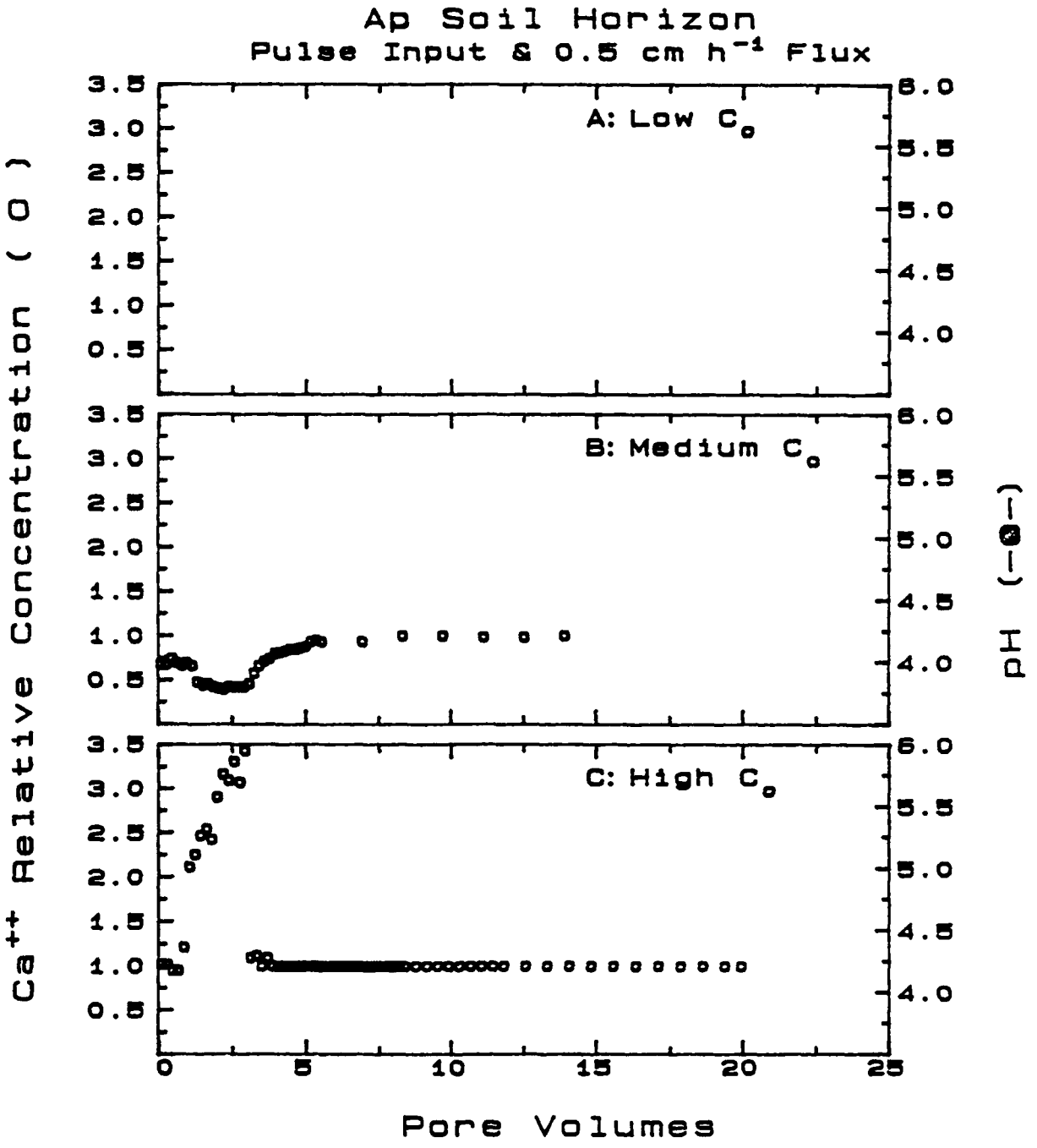


Figure 70. Effluent pH and Ca^{2+} Concentration from Columns of Ap Soil that Received Pulse Application of Influent with A. Low, B. Medium, and C. High C_o. Liquid Flux was 0.5 cm h^{-1}.

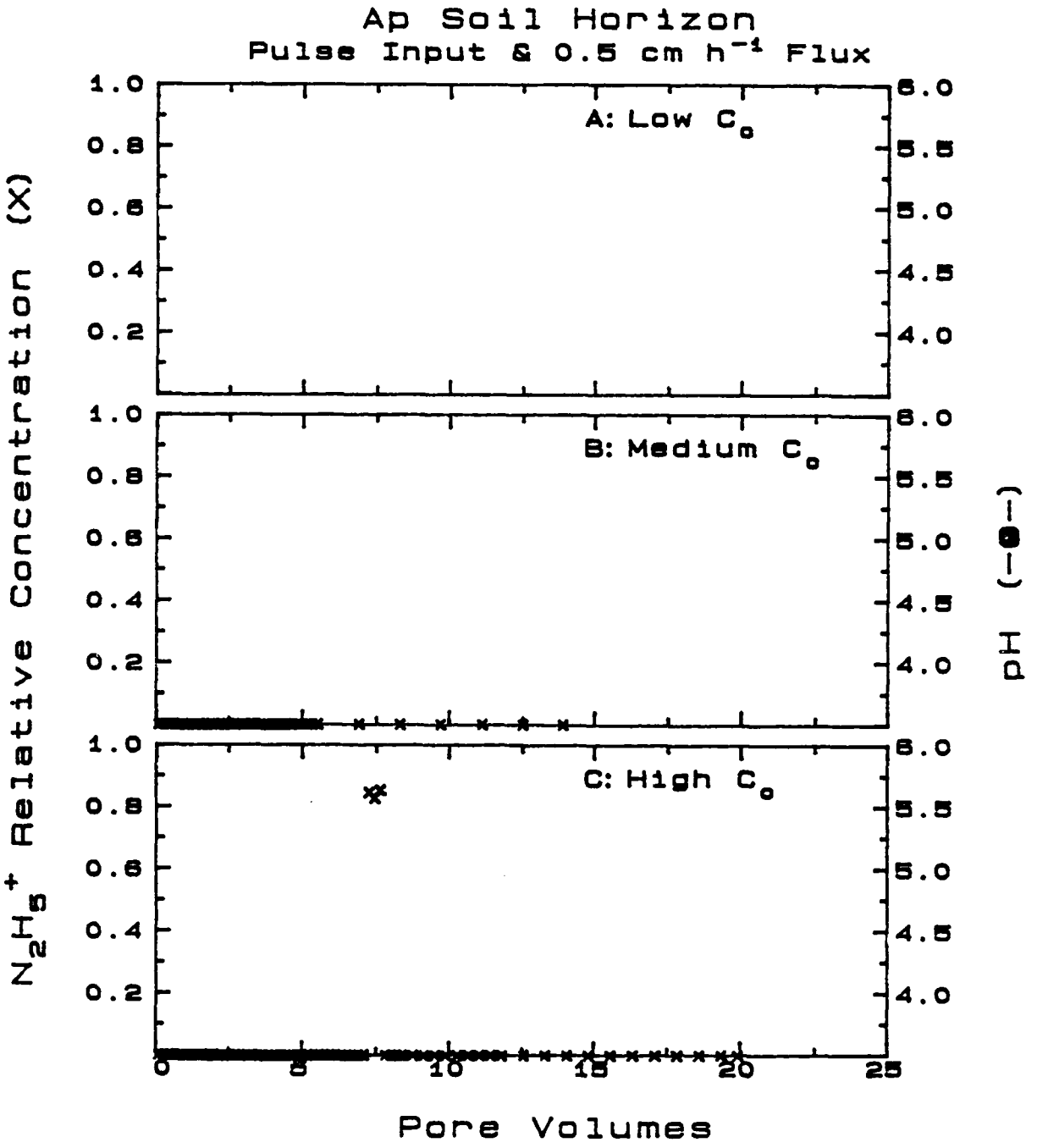


Figure 71. Effluent pH and Hydrazinium Concentration from Columns of Ap Soil that Received Pulse Application of Influent with A. Low, B. Medium, and C. High C_o. Liquid Flux was 5 cm h^{-1}.

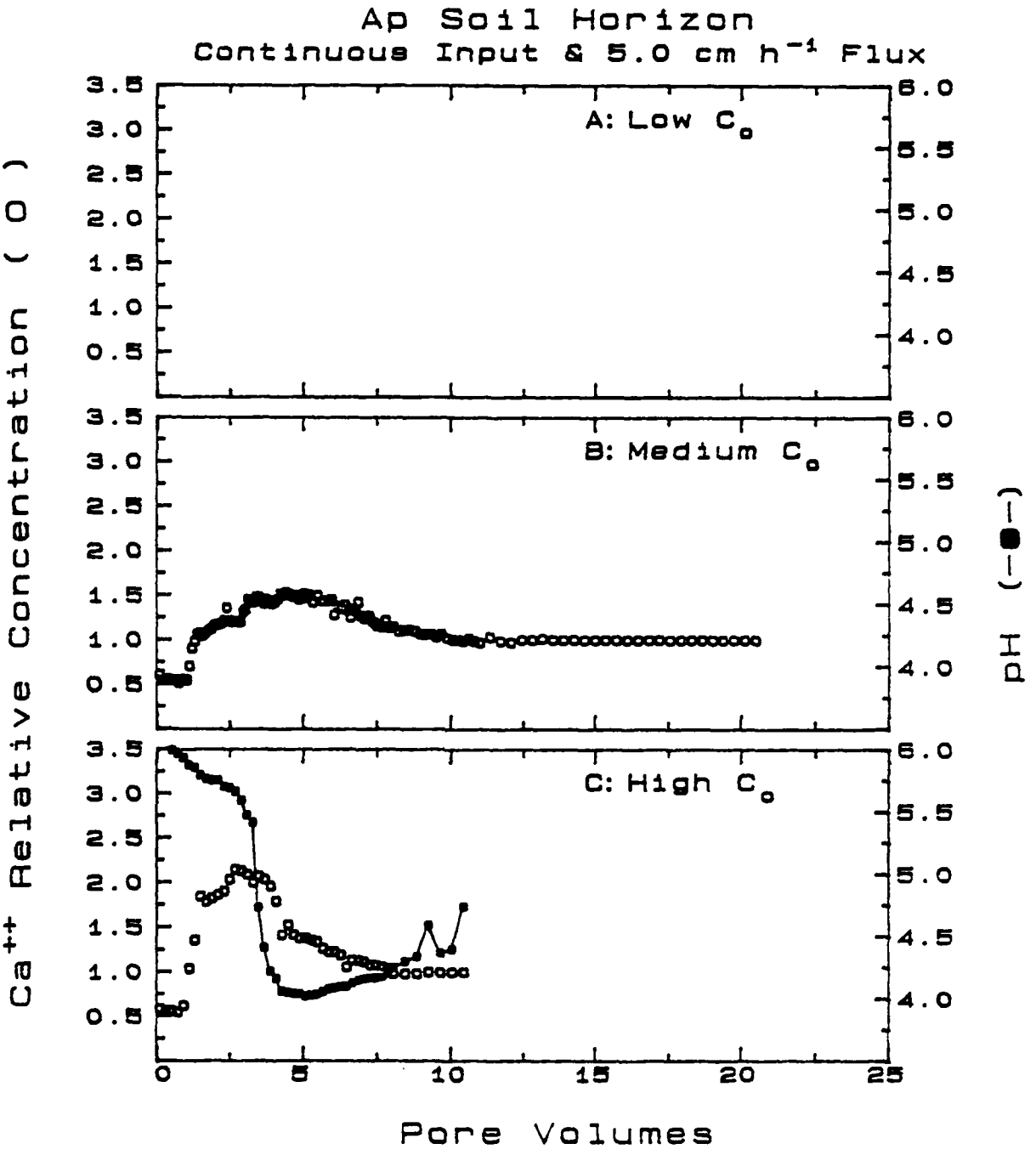


Figure 72. Effluent pH and Ca^{2+} Concentration from Columns of Ap Soil that Received Continuous Application of Influent with A. Low, B. Medium, and C. High C_o. Liquid Flux was 5 cm h^{-1}.

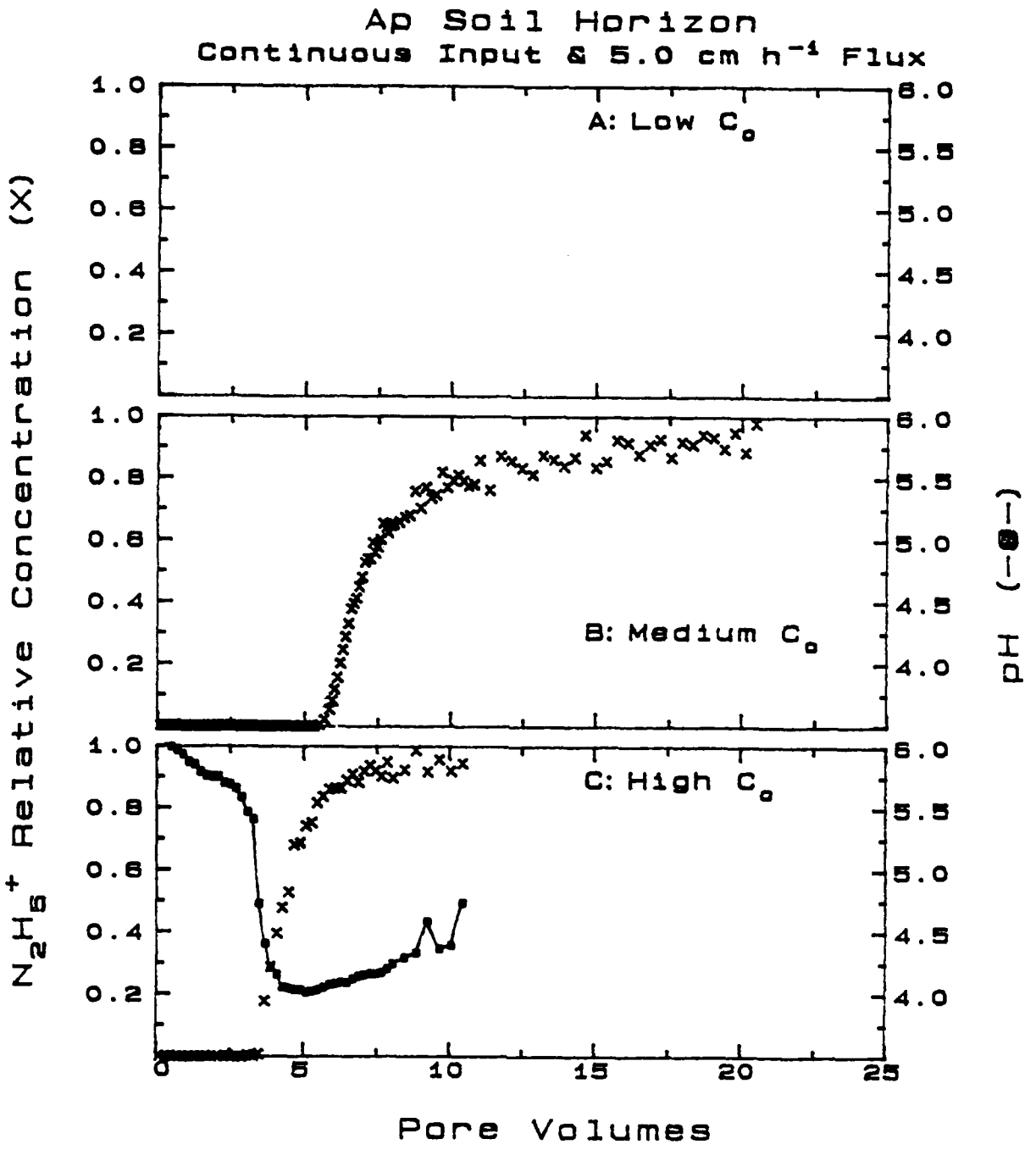


Figure 73. Effluent pH and Hydrazinium Concentration from Columns of Ap Soil that Received Continuous Application of Influent with A. Low, B. Medium, and C. High C_o. Liquid Flux was 5 cm h^{-1}.

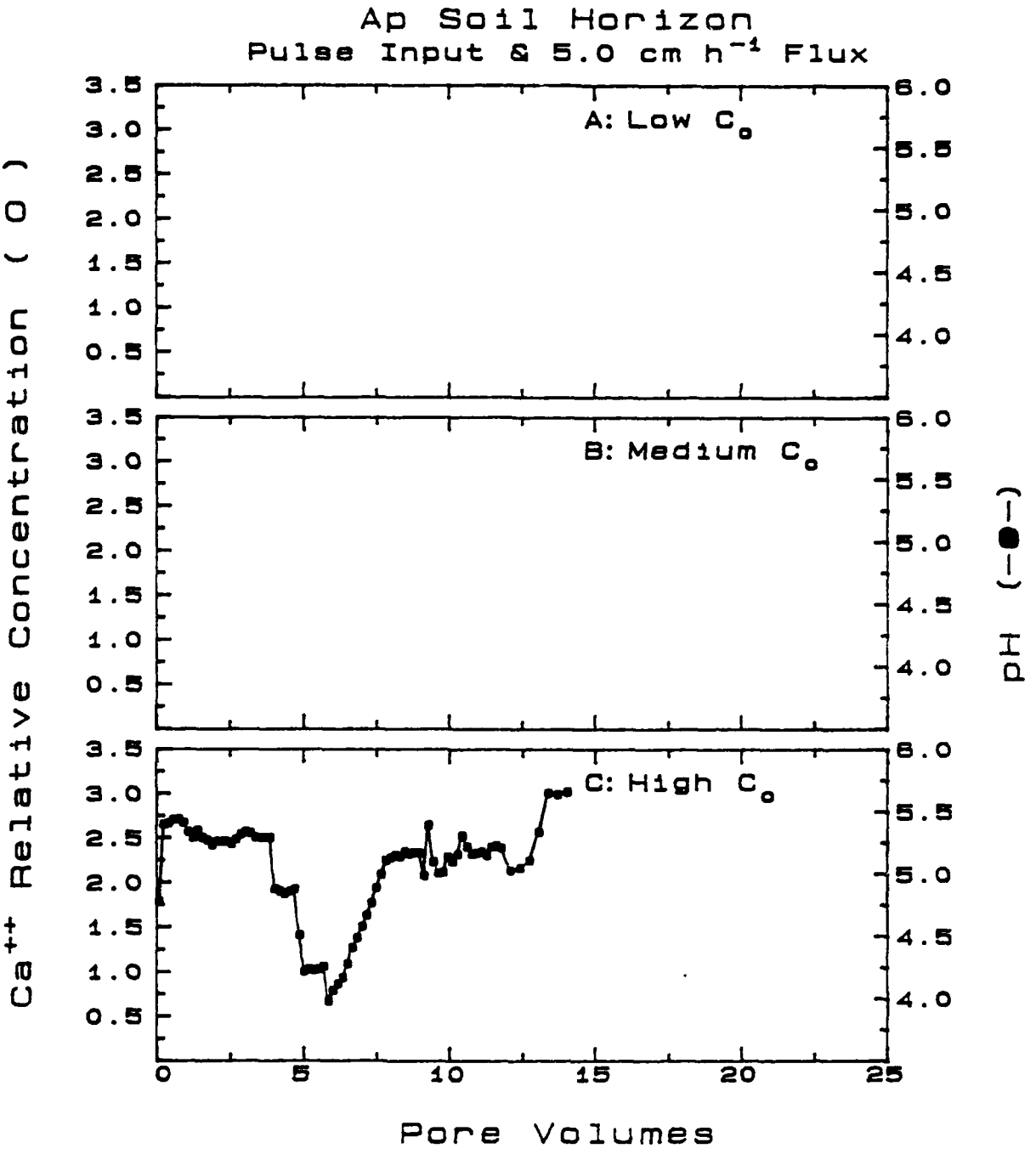


Figure 74. Effluent pH and Ca^{2+} Concentration from Columns of Ap Soil that Received Pulse Application of Influent with A. Low, B. Medium, and C. High C_o. Liquid Flux was 5 cm h^{-1}.

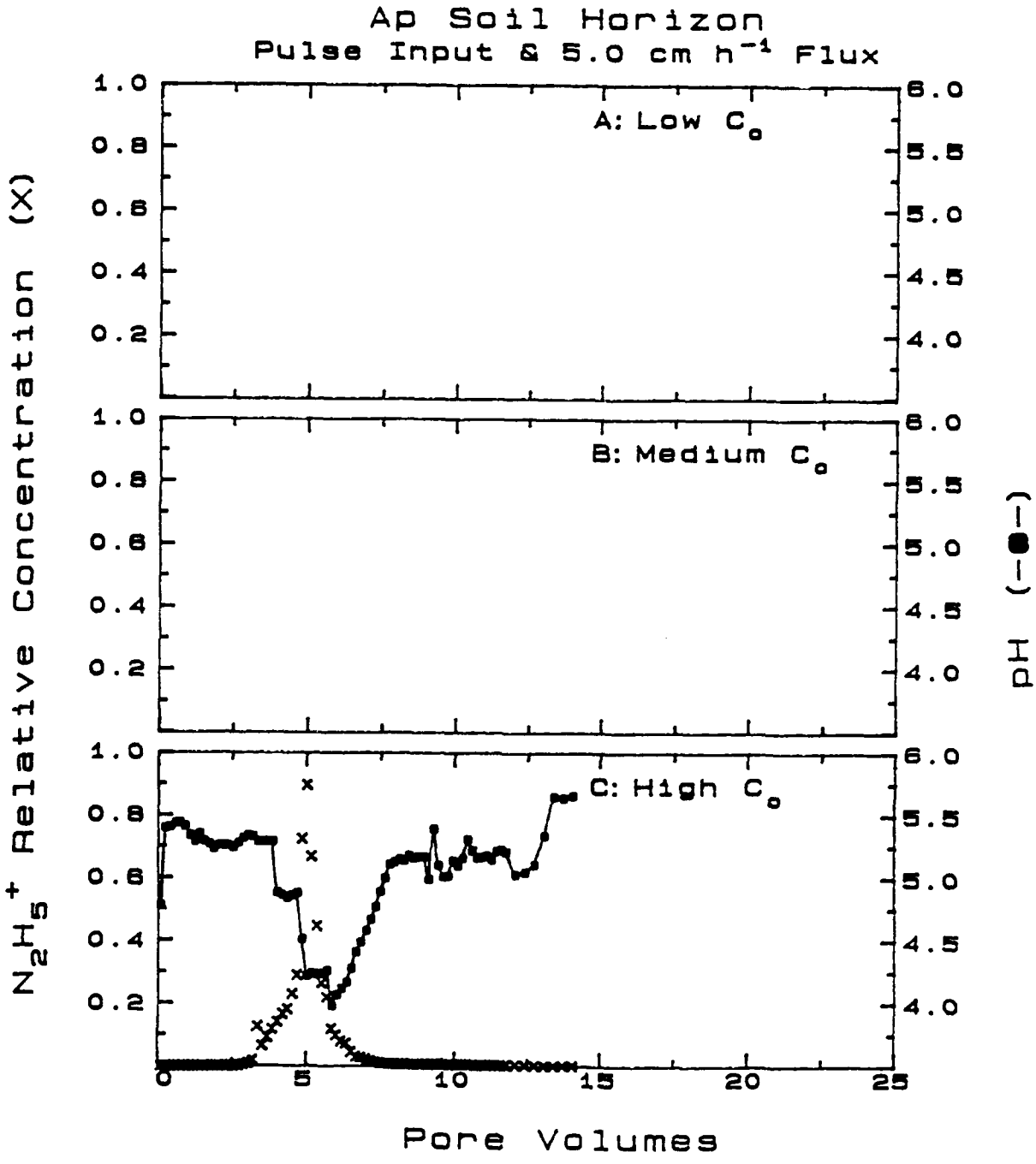


Figure 75. Effluent pH and Hydrazinium Concentration from Columns of Ap Soil that Received Pulse Application of Influent with A. Low, B. Medium, and C. High C_o. Liquid Flux was 5 cm h^{-1}.

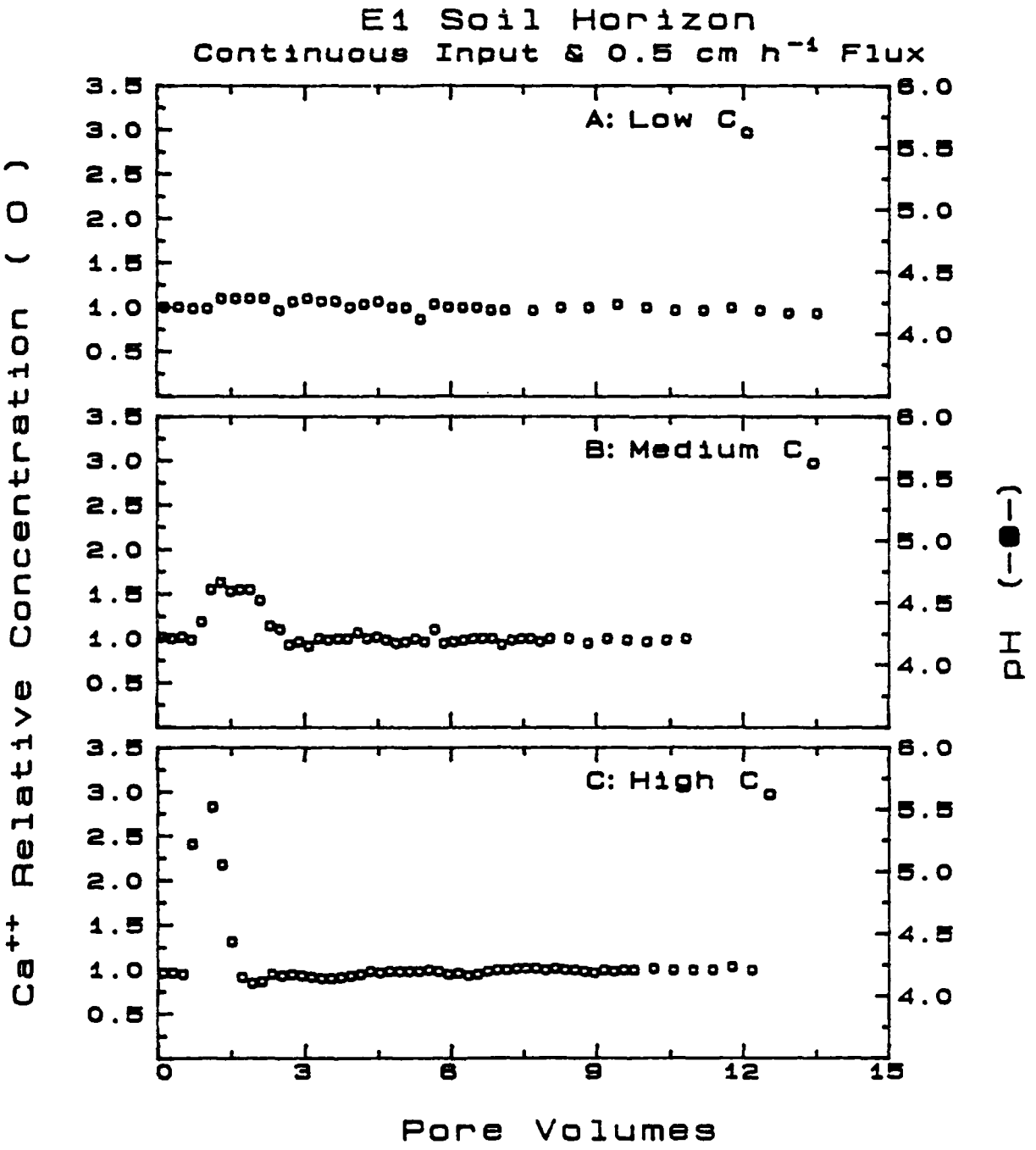


Figure 76. Effluent pH and Ca^{2+} Concentration from Columns of E1 Soil that Received Continuous Application of Influent with A. Low, B. Medium, and C. High C_o. Liquid Flux was 0.5 cm h^{-1}.

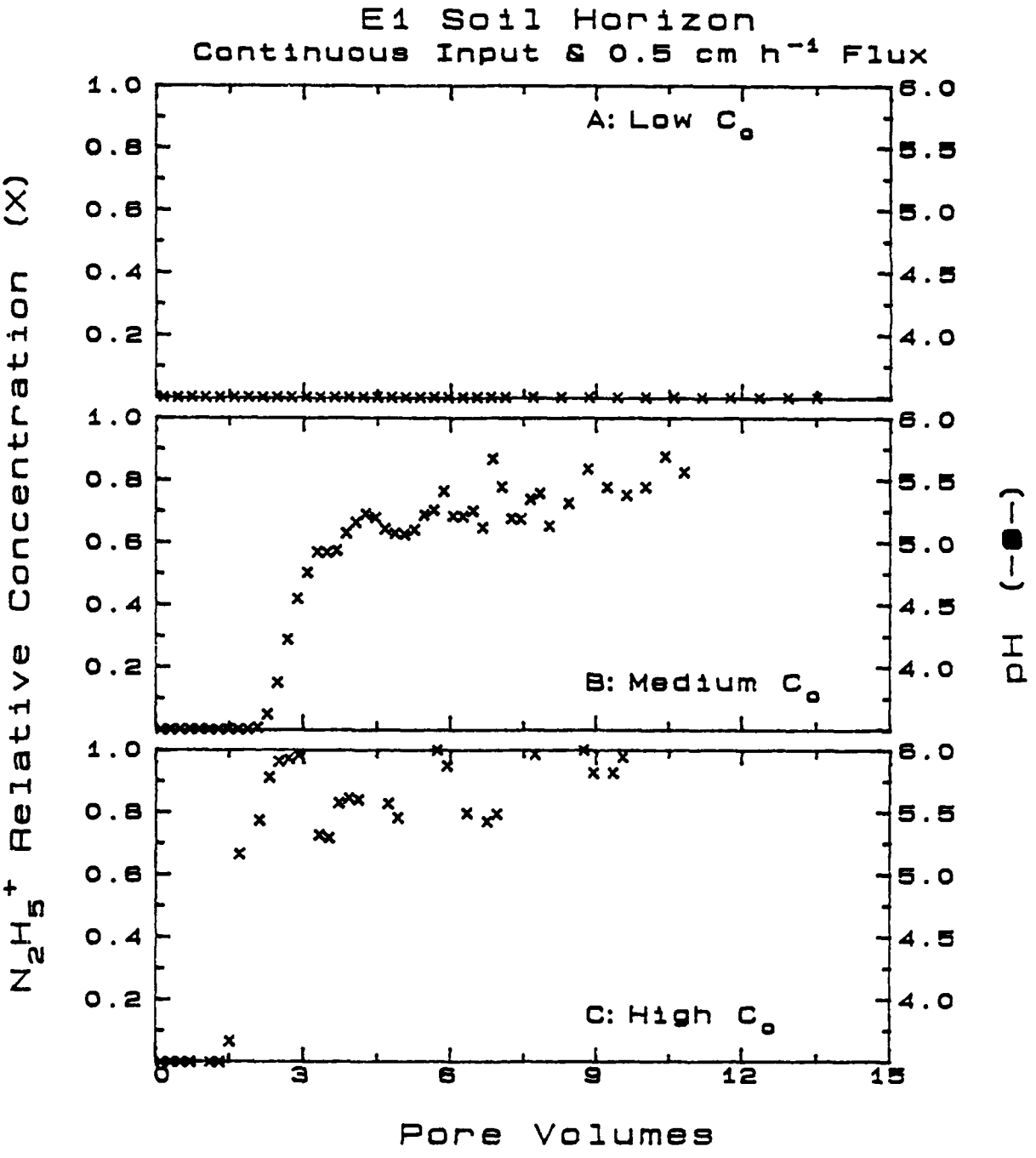


Figure 77. Effluent pH and Hydrazinium Concentration from Columns of E1 Soil that Received Continuous Application of Influent with A. Low, B. Medium, and C. High C_o. Liquid Flux was 0.5 cm h^{-1}.

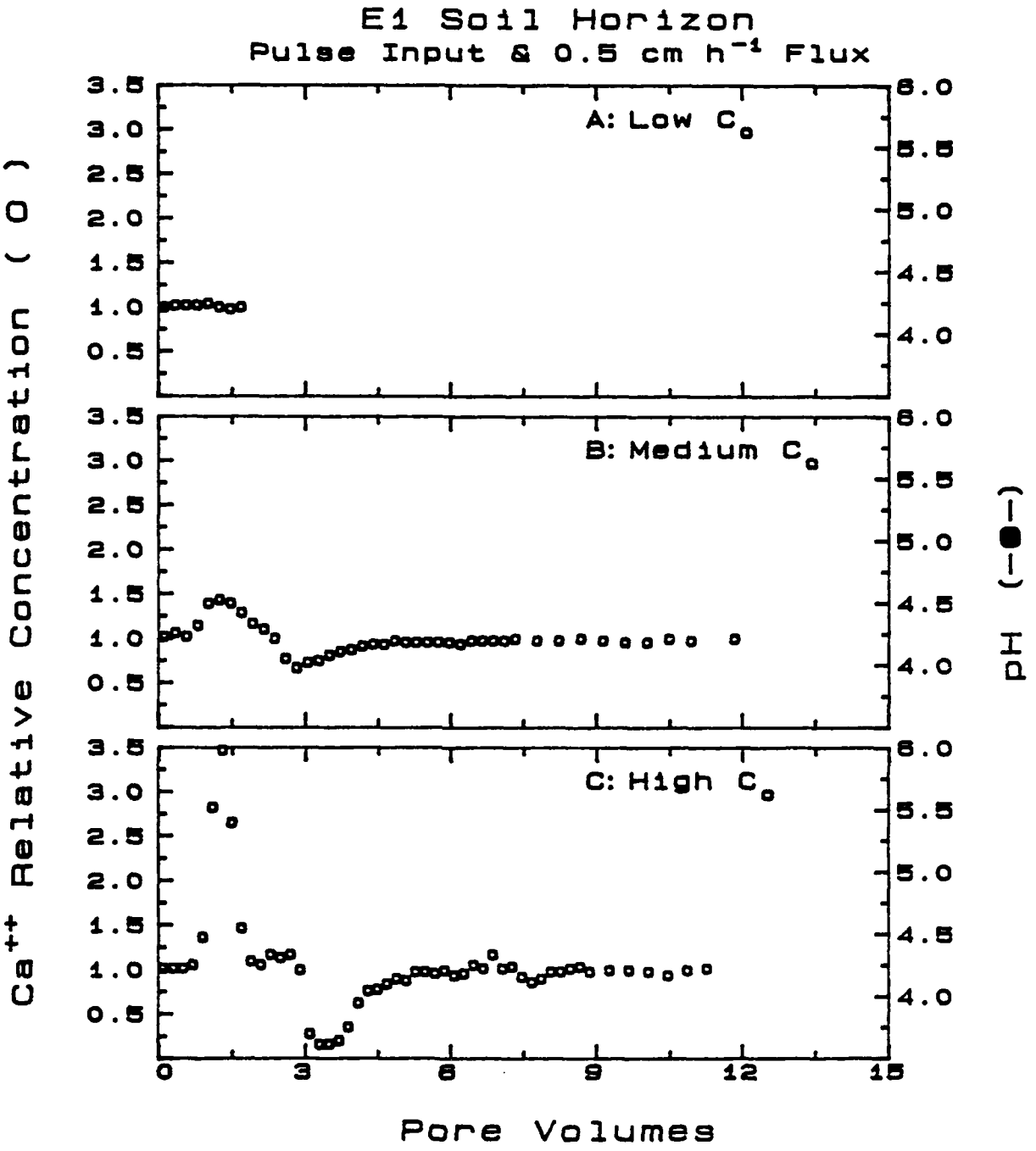


Figure 78 Effluent pH and Ca^{2+} Concentration from Columns of E1 Soil that Received Pulse Application of Influent with A. Low, B. Medium, and C. High C_o. Liquid Flux was 0.5 cm h^{-1}.

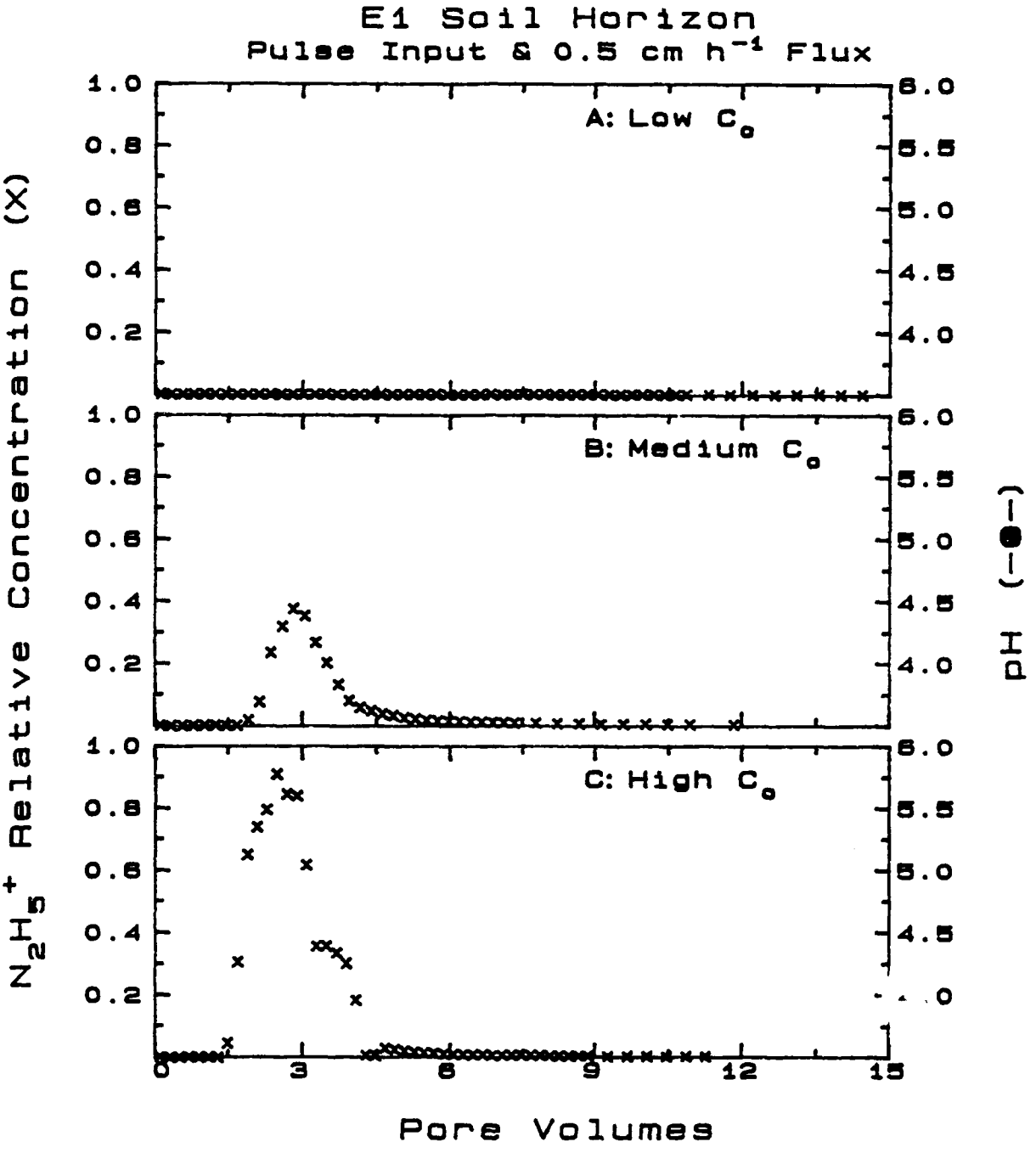


Figure 79. Effluent pH and Hydrazinium Concentration from Columns of E1 oil that Received Pulse Application of Influent with A. Low, B. Medium and C. High C_o. Liquid Flux was 0.5 cm h^{-1}.

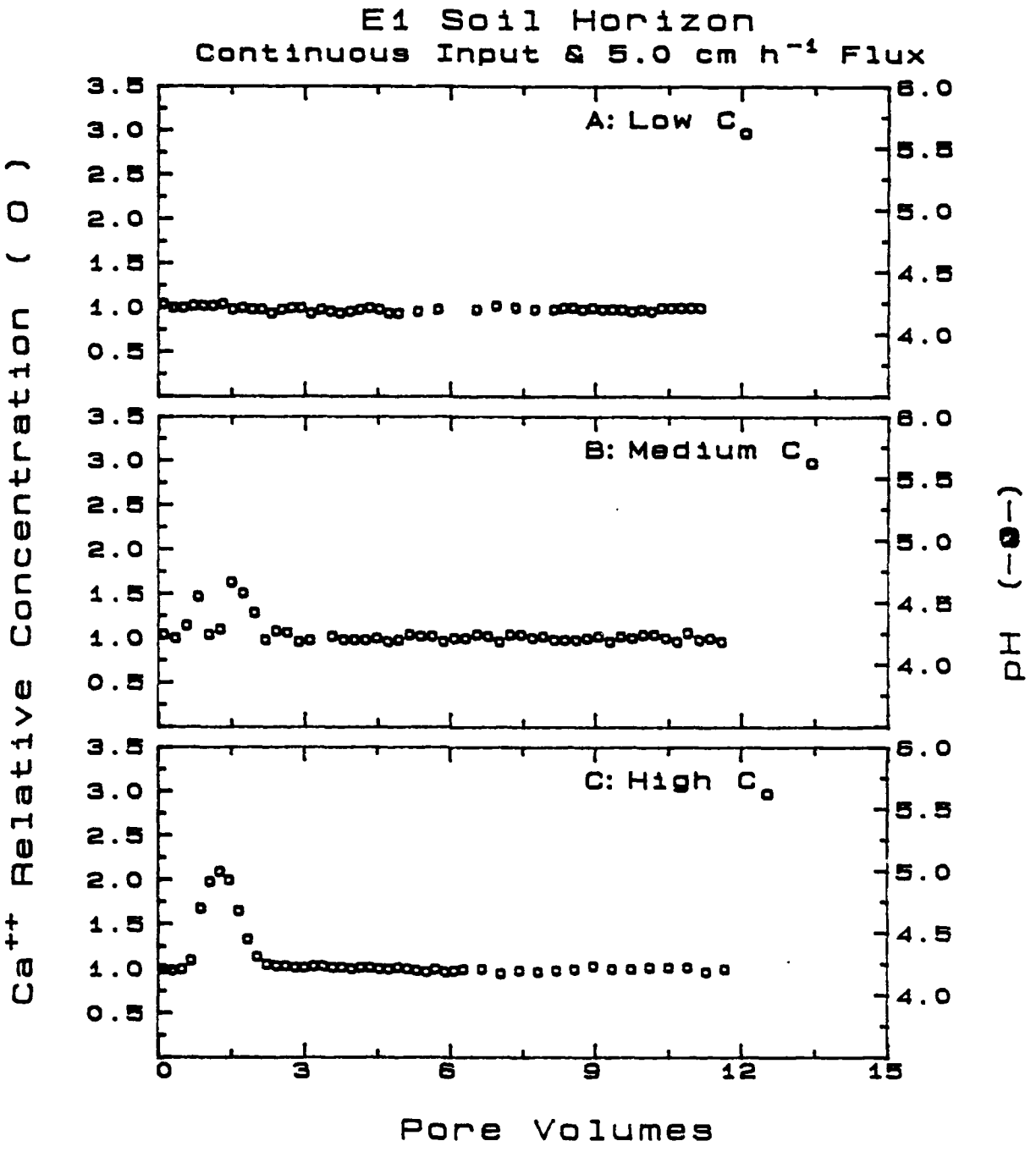


Figure 80. Effluent pH and Ca^{2+} oncentration from Columns of E1 Soil that Received Continuous Application of Influent with A. Low, B. Medium, and C. High C_o. Liquid Flux was 0.5 cm h^{-1}.

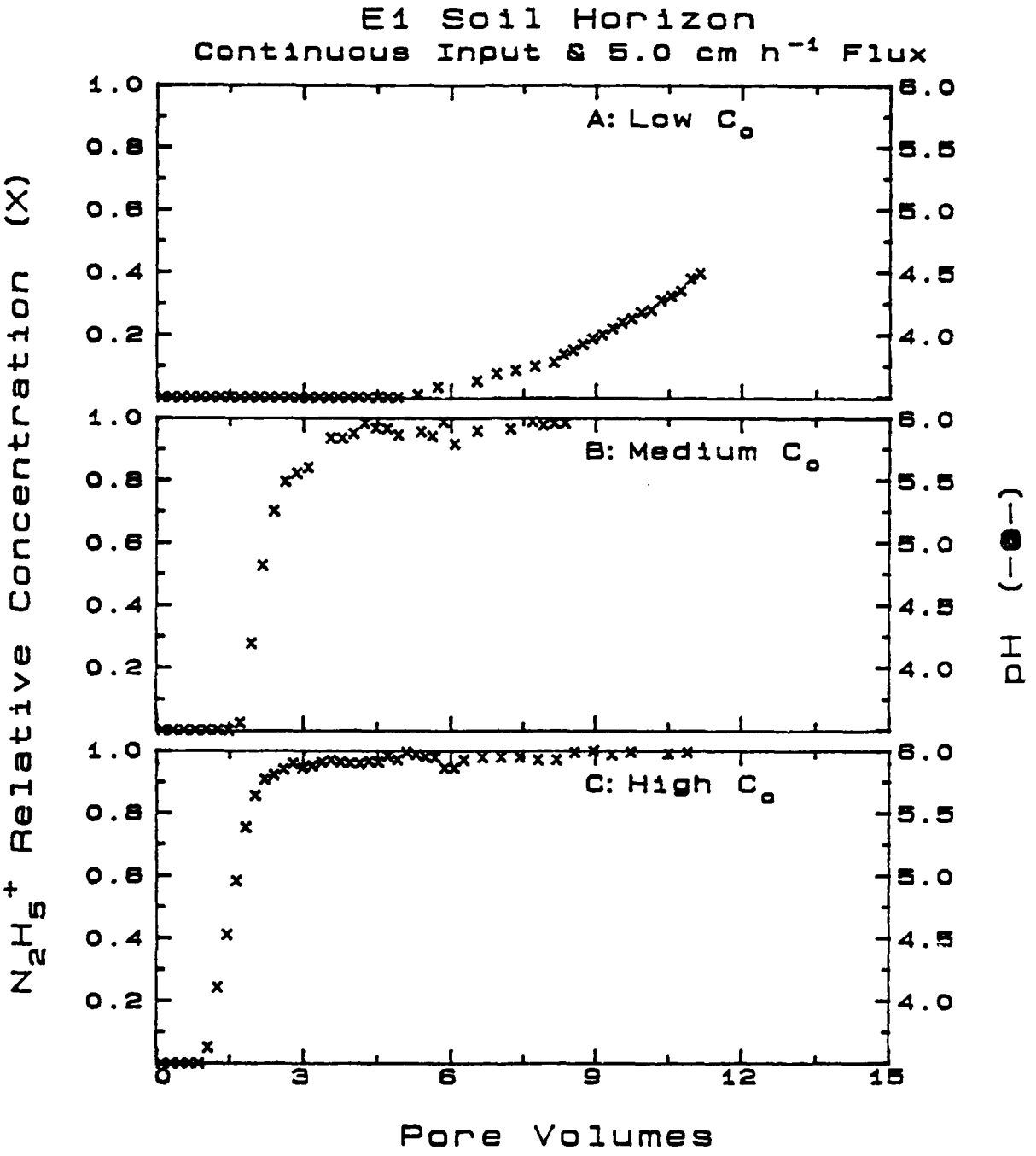


Figure 81. Effluent pH and Hydrazinium Concentration from Columns of El Soil that Received Continuous Application of Influent with A. Low, B. Medium, and C. High C_o. Liquid Flux was 5 cm h^{-1}.

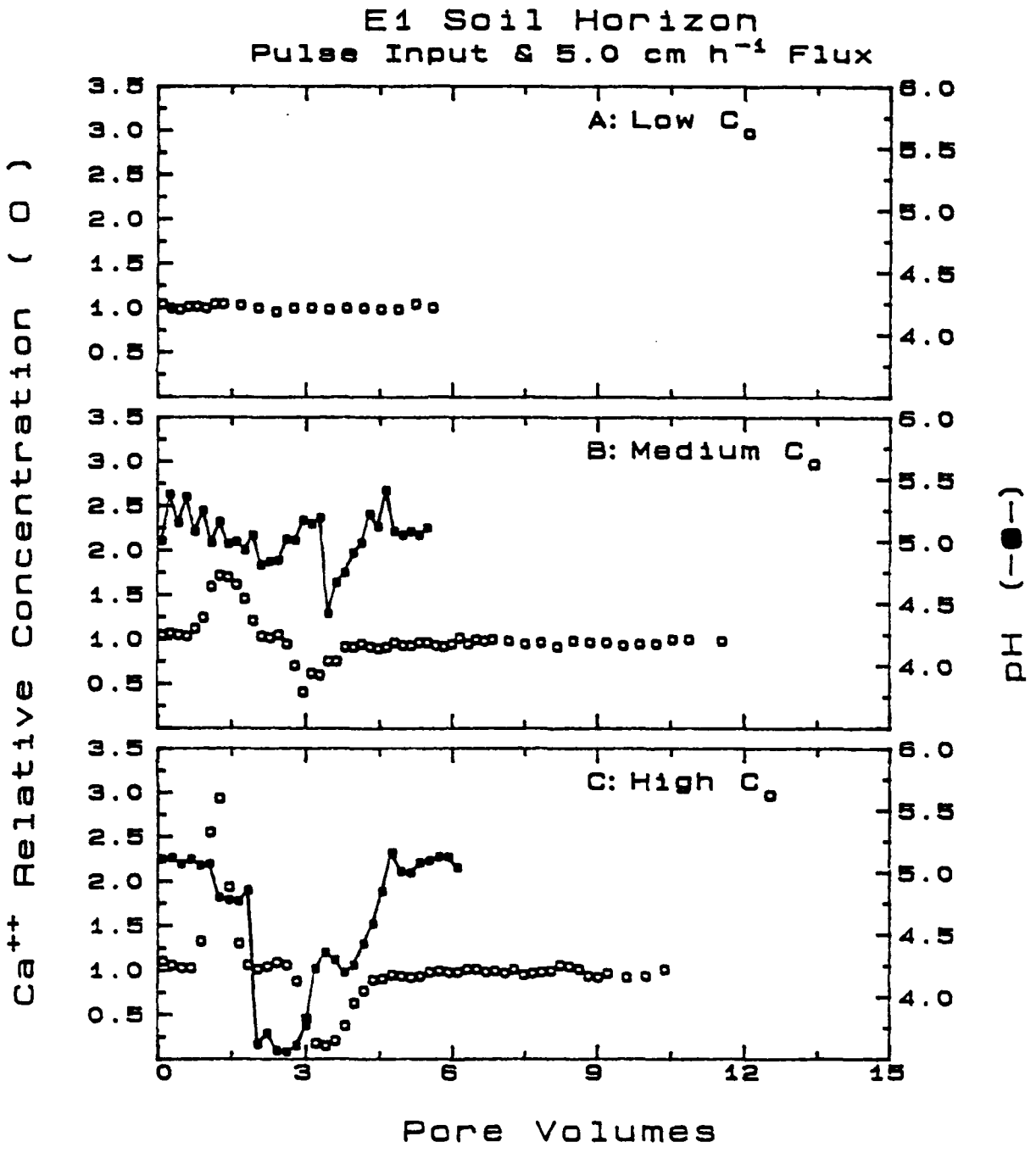


Figure 82. Effluent pH and Ca^{2+} Concentration from Columns of E1 Soil that Received Pulse Application of Influent with A. Low, B. Medium, and C. High C_o. Liquid Flux was 5 cm h^{-1}.

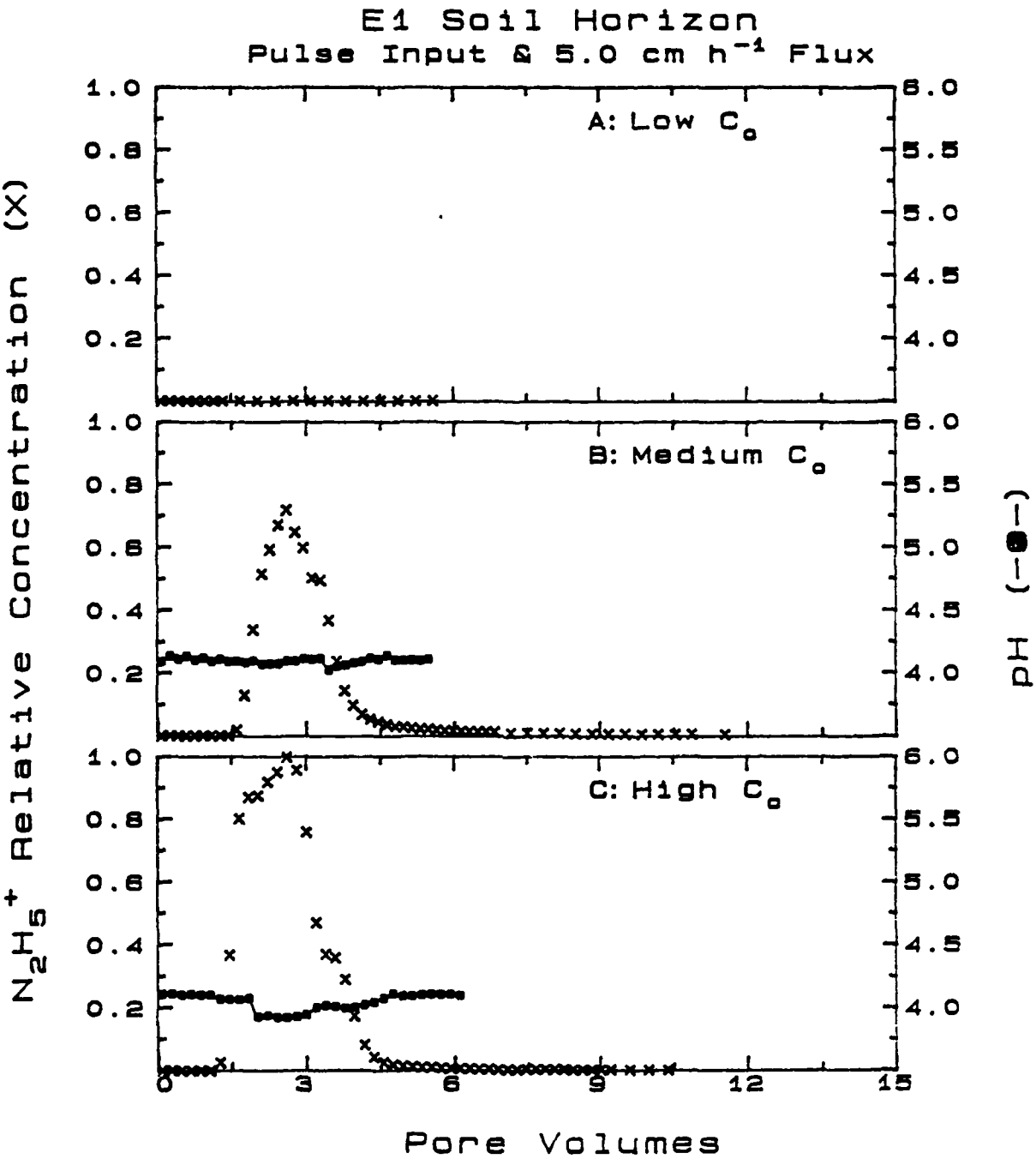


Figure 8.3. Effluent pH and Hydrazinium Concentration from Columns of E1 Soil that Received Pulse Application in Influent with A. Low, B. Medium, and C. High C_o. Liquid Flux was 5 cm h^{-1}.

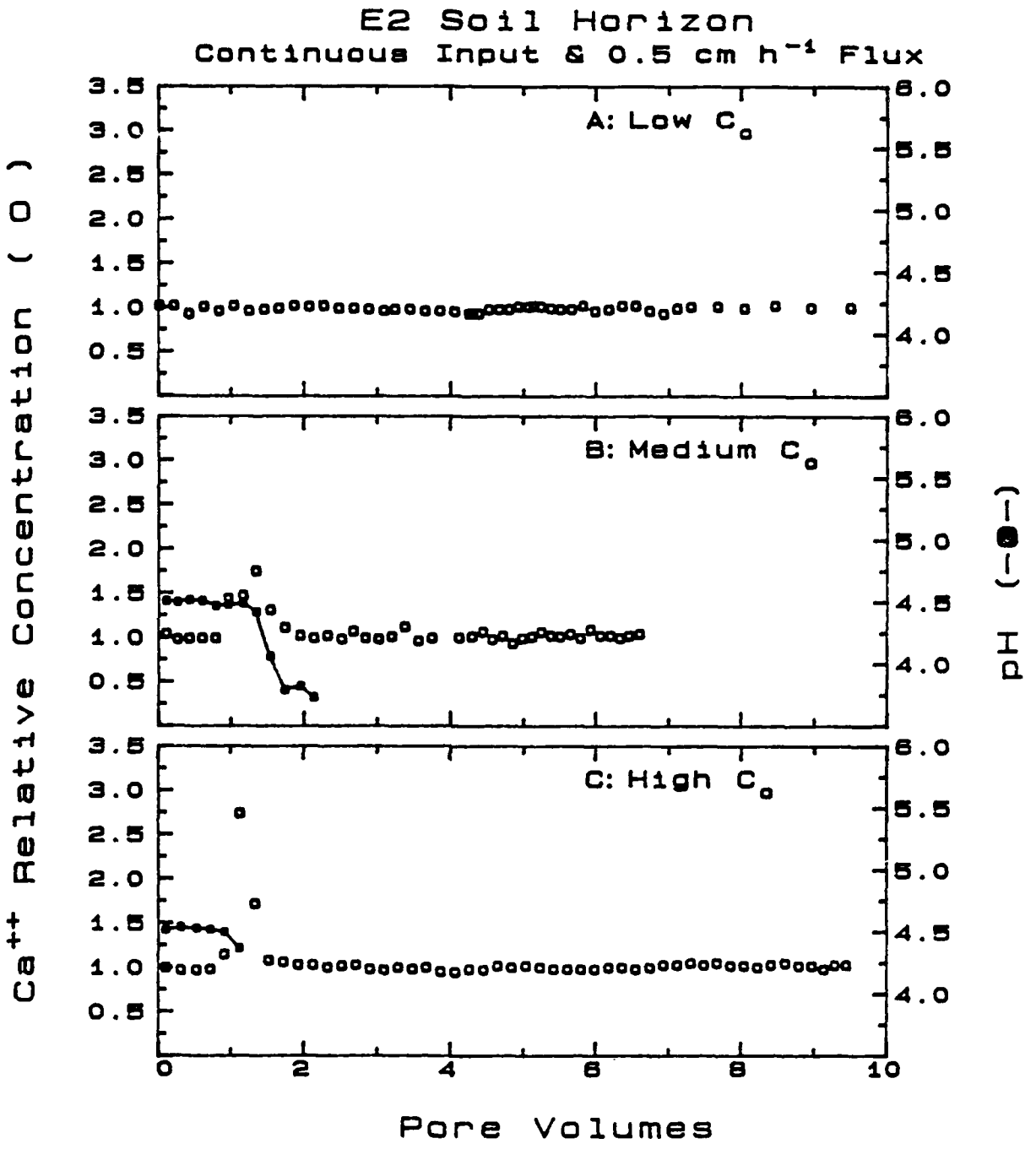


Figure 84. Effluent pH and Ca^{2+} Concentration from Columns of E2 Soil that Received Continuous Application of Influent with A. Low, B. Medium, and C. High C_o. Liquid Flux was 0.5 cm h^{-1}.

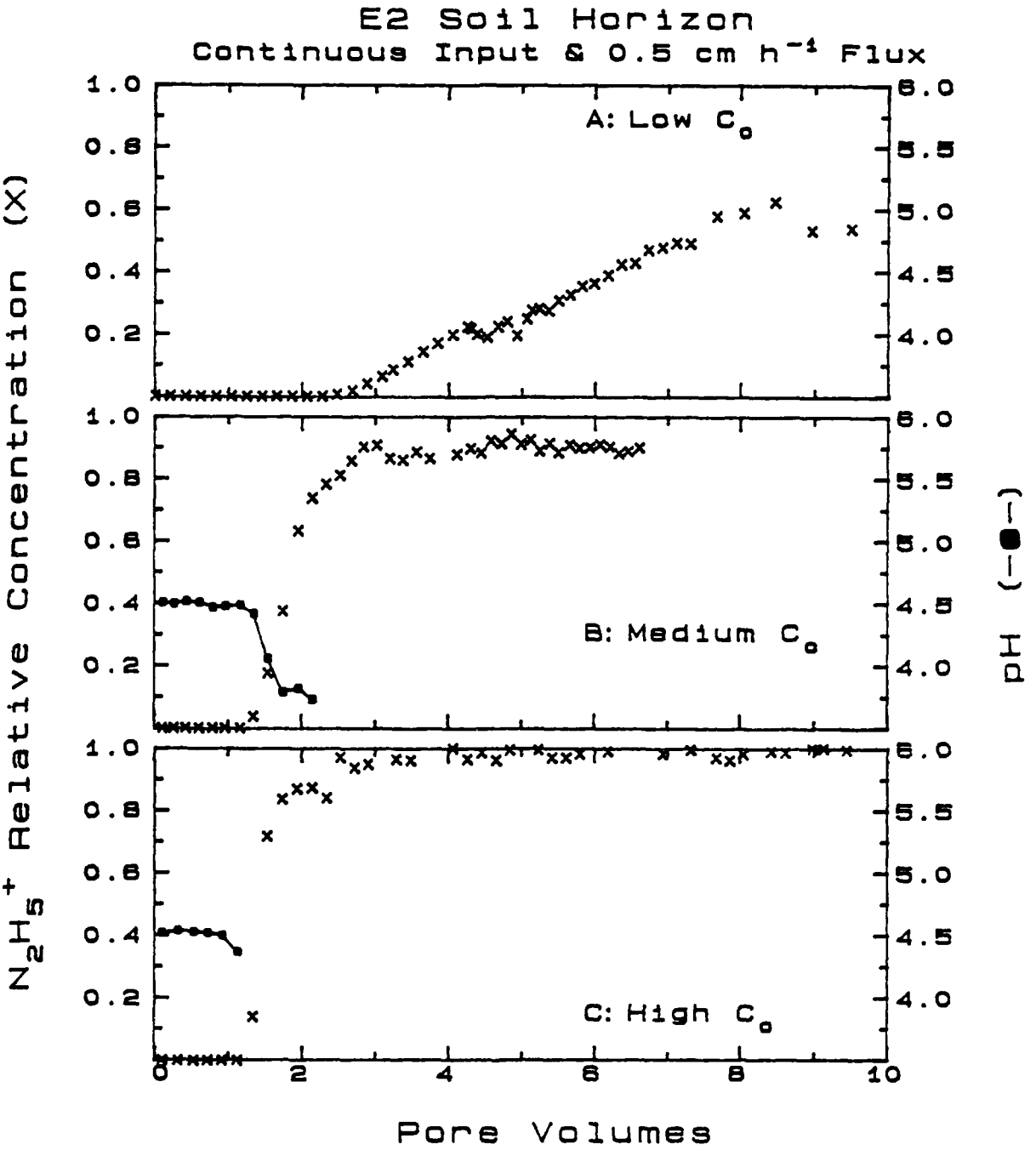


Figure 85. Effluent pH and Hydrazinium Concentration from Columns of E2 Soil that Received Continuous Application of Influent with A. Low, B. Medium, and C. High C_o. Liquid Flux was 0.5 cm h^{-1}.

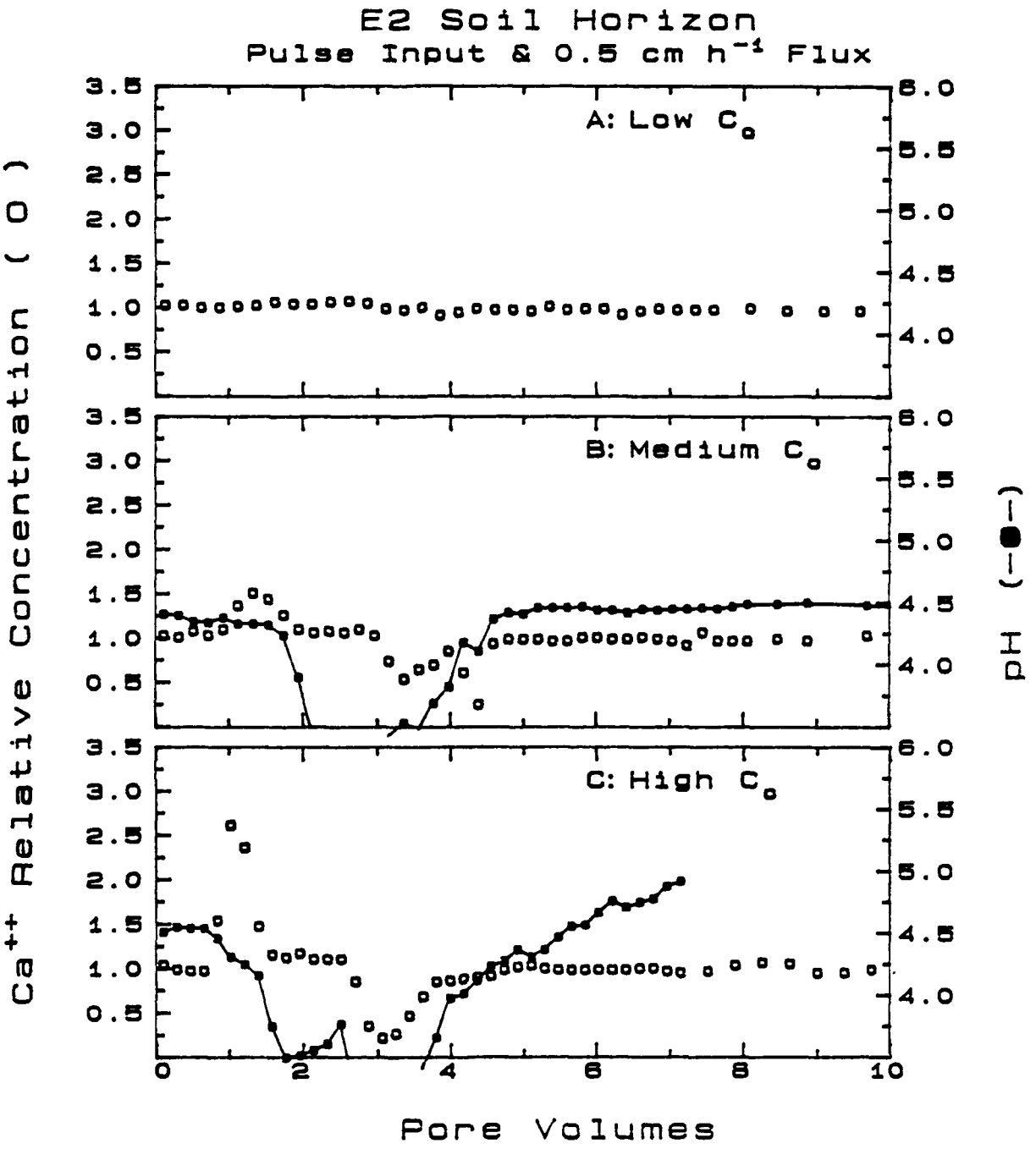


Figure 86. Effluent pH and Ca^{2+} Concentration from Columns of E2 Soil that Received Pulse Application of Influent with A. Low, B. Medium, and C. High C_o. Liquid Flux was 0.5 cm h^{-1}.

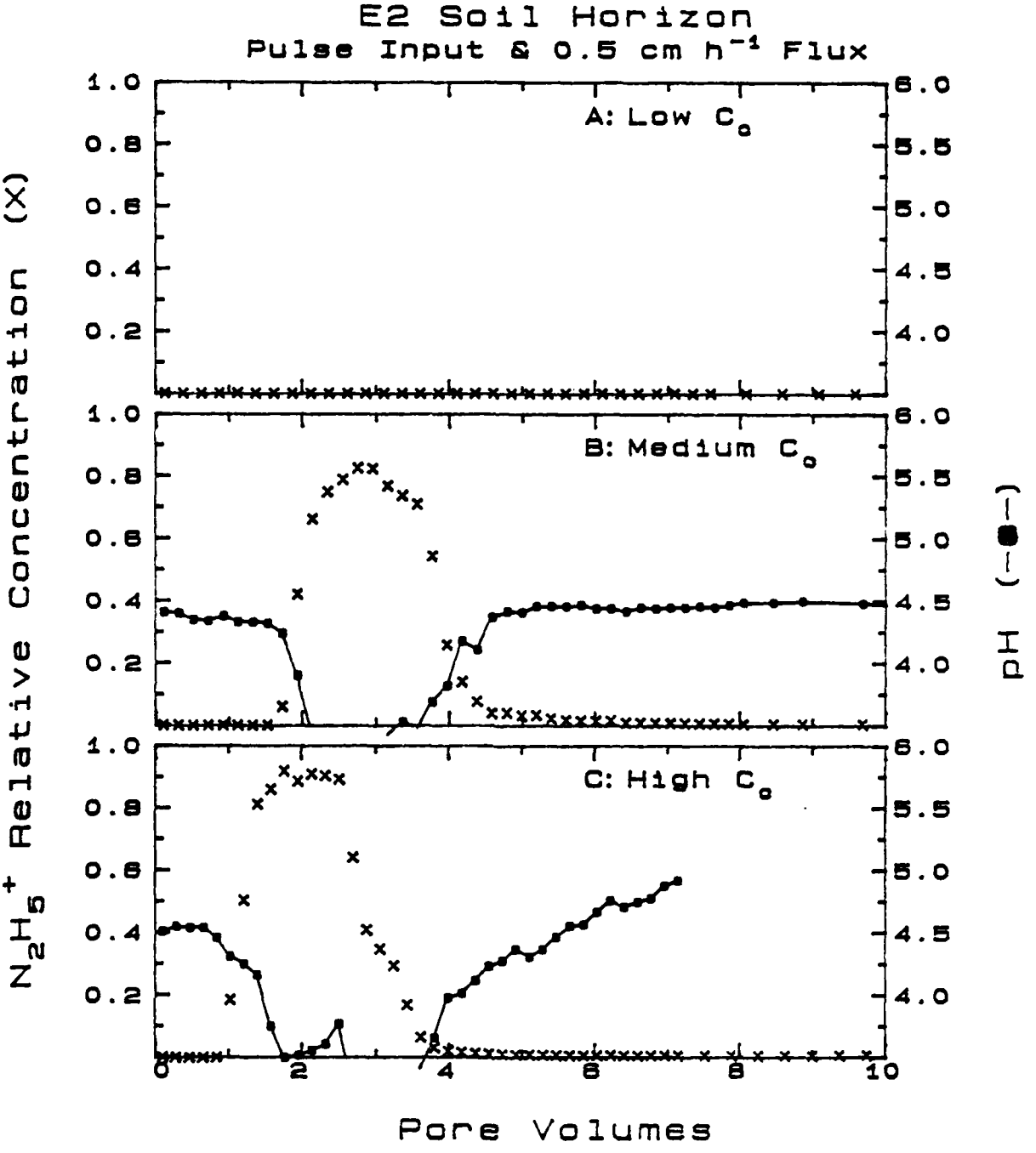


Figure 87. Effluent pH and Hydrazinium Concentration from Columns of E2 Soil that Received Pulse Application of Influent with A. Low, B. Medium, and C. High C_o. Liquid Flux was 0.5 cm h^{-1}.

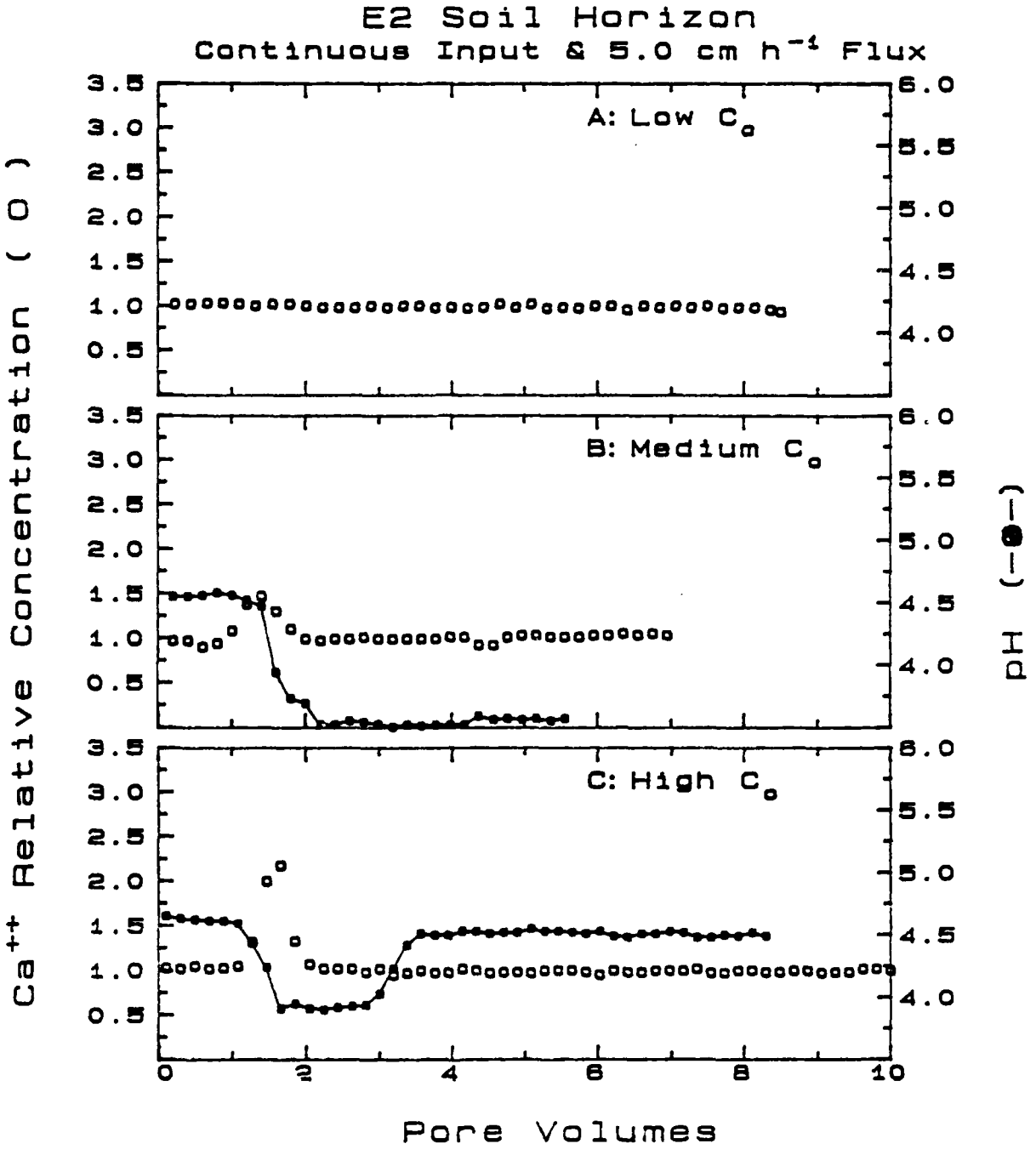


Figure 88. Effluent pH and Ca^{2+} Concentration from Columns of E2 Soil that Received Continuous Application of Influent with A. Low, B. Medium, and C. High C_o. Liquid Flux was 5 cm h^{-1}.

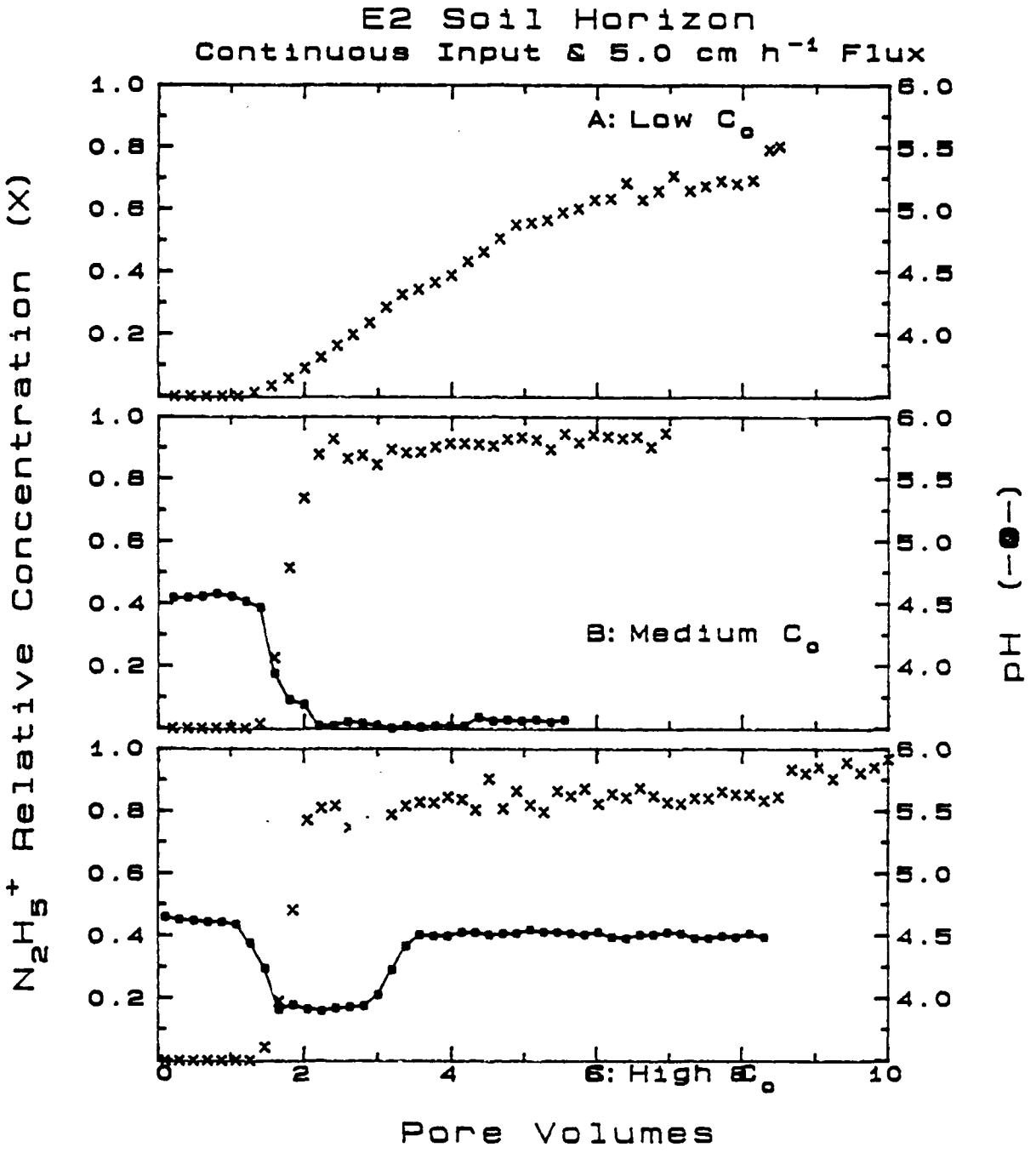


Figure 89. Effluent pH and Hydrazinium Concentration from Columns of E2 Soil that Received Continuous Application of Influent with A. Low, B. Medium, and C. High C_o. Liquid Flux was 5 cm h^{-1}.

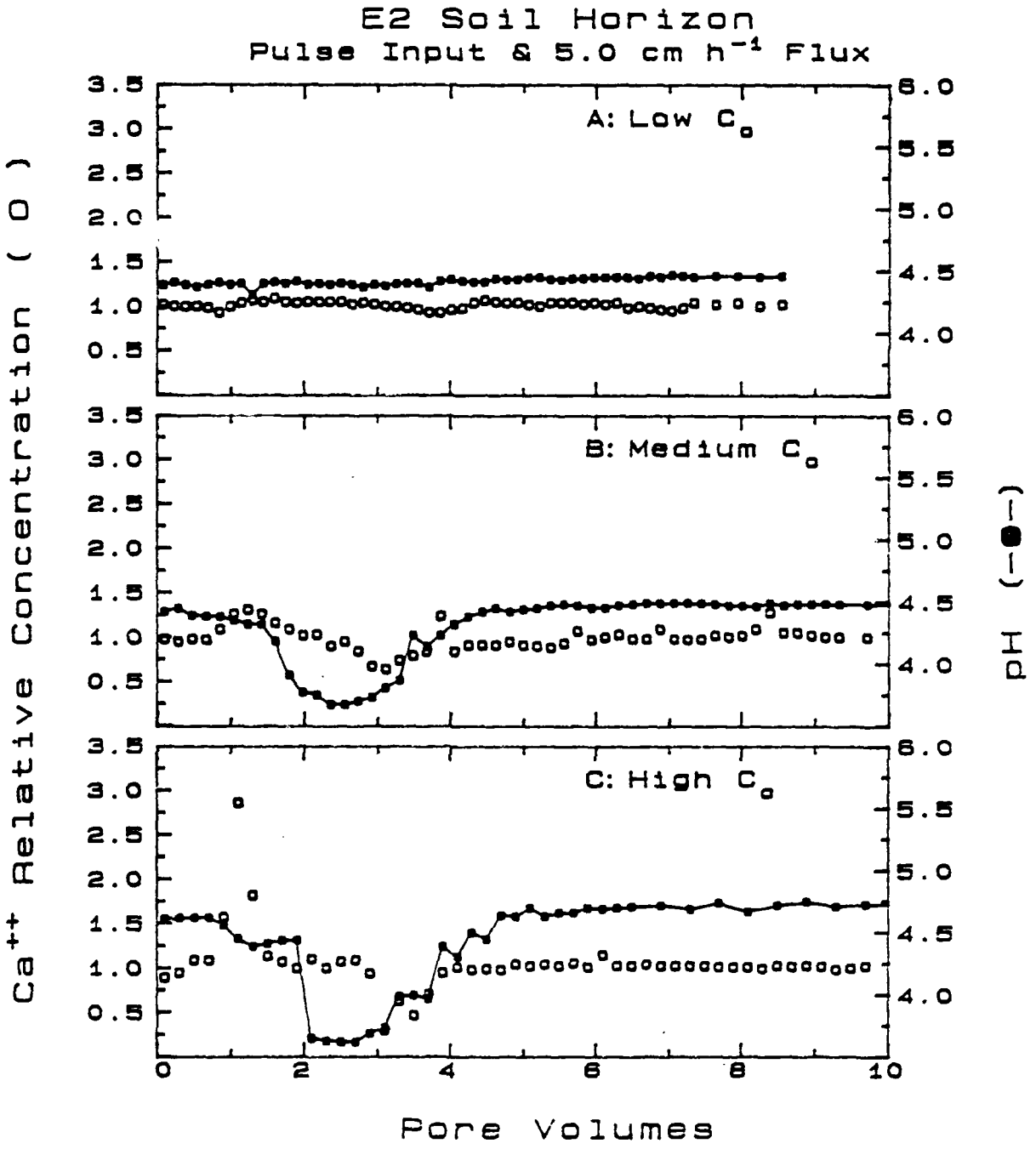


Figure 90. Effluent pH and Ca^{2+} Concentration from Columns of E2 Soil that Received Pulse Application of Influent with A. Low, B. Medium, and C. High C_o. Liquid Flux was 5 cm h^{-1}.

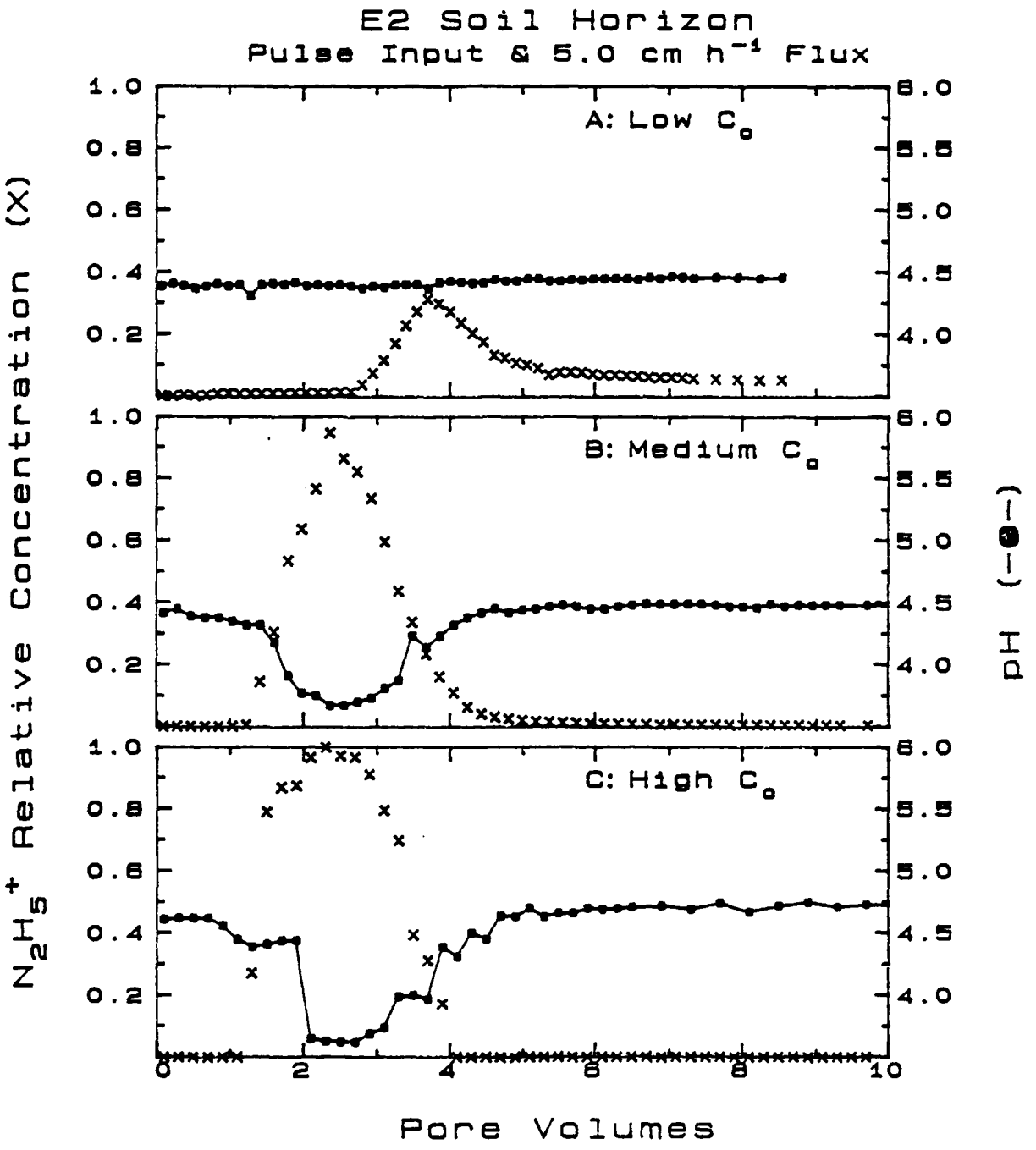


Figure 91. Effluent pH and Hydrazinium Concentration from Columns of E2 Soil that Received Pulse Application of Influent with A. Low, B. Medium, and C. High C_o. Liquid Flux was 5 cm h^{-1}.

APPENDIX B

COMPUTER CODE FOR ONE-DIMENSIONAL MODEL

1. Computer Code for One-Dimensional Model

Program for preparation of parameter files for the hydrazine simulation program:

SETHYDRA Variable Dictionary

```
AmBnpH          = ambient pH
AutoXB          = Boolean indicating whether auto-oxidation is to occur
BasePH          = base pH (used with ambient pH in hydrolysis calculations)
BulkDn          = bulk density (gm/cc)
Column          = column length (cm)
Conc(MaxSet,6)  = concentration of dissolved N2H4, dissolved & sorbed N2H5+, dissolved & sorbed Ca++, and
                  dissolved oxygen. Dissolved values are in mew/cc and sorbed values in meq/gm.  There can
                  be up to MAXSET sets of values, each corresponding to a boundary condition in the column
Depth(Maxset)   = the depth at which each of the boundary condition sets applies
Disper          = dispersion coefficient (cm2/sec)
DtMax           = maximum value that Dt (the time step) will be allowed to assume (sec)
DtStep          = the increment (fractional) that Dt will be increased from Dtmin to DtMax
Dx              = the increment between nodes (cm)
EffZer          = effective zero exponent (typically -30 --> 1.0E-30)
Feedbk          = boolean indicating feedback for debugging purposes
FlConc(MaxSet,4) = the concentration of N2H4,N2H5+,Ca++ and O2 in the input flux for MAXSET possible flux changes
FluxAm(MaxSet)  = the flux (cm/sec) imposed on the column at any given flux change
FluxTm(MaxSet)  = when the flux change occurs (secs)
Fprint          = filename used to receive the simulation report
HydroB          = boolean indicating whether hydrolysis is to occur
Icheck          = the number of iterations of the simulation between checks on the abort function
IonExB          = boolean indicating whether hydrolysis is to occur
IrrevB          = boolean indicating whether irreversible sorption is to occur
ItrLim          = iterative limit - the number of trys to generate a convergent solution
Kinetc(2)       = boolean indicating whether a kinetic approach for N2H4 and N2H5+ reversible sorption is to be used
MaxSet= 5       = maximum number of boundary condition sets and flux changes allowed
MicroB          = boolean indicating whether microbial degradation is to occur
Nflux           = the number of flux changes
Nreps           = the number of reports requested
Nspecs          = the number of boundary condition sets needed
RautoX(2,3)     = the rate coefficients for auto-oxidation
Rdgrad(2,4)     = the rate coefficients for microbial degradation
RepTim(MaxSet)  = the times at which reports are to be generated
RevspB          = boolean indicating whether reversible sorption is to occur
Rlabel          = the overall run label (80 characters)
Rsorb(2,8)      = the reversible sorption coefficients for (N2H4 & N2H5+) by (kf,kb,N,kff,kpb,kp,kq). The
                  last coefficient (.,8) is the irreversible 1st order degradation coefficient
Scale(4)        = divisors for the four species by which the breakthrough curve values are scaled respectively
SelH2C          = selectivity coefficient between calcium and N2H5+
Sponge          = maximum permanent binding capacity at each node (meq N2H5+/gm)
ThetaS          = saturated water content
TLimit          = tolerance level (%) at which the difference between successive approximations is considered
                  to be acceptable
TmStop          = time at which simulation is to be terminated (sec)
```

SETHYDRA Common Block

```
C Common blocks for SetHyd = Sethydra.CMB
      Parameter (MaxSet=5)
      Common/Info/AutoXB,AmBnpH,BasePH,BulkDn,Column,Conc(MaxSet,6),
     &            Depth(Maxset),Disper,DtStep,DtMax,Dx,EffZer,Feedbk,
     &            FlConc(MaxSet,4),FluxAm(MaxSet),FluxTm(MaxSet),
     &            HydroB,Icheck,IonExB,IrrevB,ItrLim,Kinetc(2),MicroB,
     &            Nflux,Nreps,Nspecs,RautoX(2,3),Rdgrad(2,4),
     &            RepTim(MaxSet),RevspB,Rsorb(2,8),Scale(4),SelH2C,
     &            ThetaS,TLimit,TmStop,Sponge
      Common/ChrBlk/Rlabel,Fprint
      Character Rlabel*80,Fprint*50
      Logical AutoXB,Feedbk,HydroB,IonExB,IrrevB,Kinetc,MicroB,RevspB
```

SETHYDRA Program Code

```
      PROGRAM SetHyd
C-----------------------------------------------------------------------------------------------------------------------C
C                   Dr. Stephen A. Bloom - for Dr. Robert Mansell, Soil Science 04/25/86                                C
C                   program last modified - February 28,1988
C-----------------------------------------------------------------------------------------------------------------------C
      INCLUDE 'SetHydra.CMB'
      Dimension Option(13)
      Logical Abort,AskQus,SetGen,StSoil
      Character NumStr*2,Option*80
      Integer CHoice
      Data LenOpt/63/,Nitems/13/,Option/
     & '<0>  Terminate program',' ',
     & '<1>  Set general simulation parameters (L,Dx,Dt...)',
     & '<2>  Set the soil characteristics',
     & '<3>  Set the boundary conditions (xx currently set)',
     & '<4>  Set the imposed flux & concentration conditions (xx set)',
     & '<5>  Specify when reports are to be generated (xx set)',
     & '<6>  Alter miscellaneous default parameters',
```

```
     & '<->  Record simulation parameters',
     & '<8>  Reset for entering another parameter set',
     & '<9>  Read a parameter file into memory',
     & ' ','* = Still needs to be set before saving parameters'/
      Call Page('Hydrazin(e)(ium) Transport for Columns Preparation')
10   SetGen = .false.                                    {indicate that the general and soil parameters}
     StSoil = .false.                                    {are not yet set and set the counters to zero}
     Nspecs = 0
     Nreps = 0
     Nflux = 0
     Call SetUp                                          {initialize the variables}
20   Call In2Chr(Nspecs,NumStr)                          {convert the counters to string variables}
       Do 30 I = 3,7
30       Option(I)(4:4) = ' '
       If (.Not.SetGen) Option(3)(4:4) = '*'             {flag the options if they need set}
       If (.not.StSoil) Option(4)(4:4) = '*'
       If (Nspecs.LE.0) Option(5)(4:4) = '*'
       If (Nflux.LE.0) Option(6)(4:4) = '*'
       If (Nreps.LE.0) Option(7)(4:4) = '*'
       Option(5)(35:36) = NumStr
       Call In2Chr(Nflux,NumStr)
       Option(6)(55:56) = NumStr
       Call In2Chr(Nreps,NumStr)
       Option(7)(48:49) = NumStr
       Choice = MenChc(LenOpt,Nitems,Option,.true.)      {display the menu and get the users choice}
       If (CHoice.LE.0) THEN
         If (.Not.AskQus('Do you wish to terminate?',.true.,12))   {confirm termination before termination}
     &       Goto 20
       ELSE IF (Choice.EQ.1) THEN
         Call RunPrm(SetGen)                                       {set the general run parameters}
       ELSE IF (Choice.EQ.2) THEN
         Call SoilVl(StSoil)                             {set the soil values}
       ELSE IF (Choice.EQ.3) THEN
         If (.Not.SetGen) THEN
           Call Warnin('Please set the general conditions first',
     &         .true.,12)
         ELSE
           Call Bounds                                   {set the boundary conditions}
         ENDIF
       ELSE IF (Choice.EQ.4) THEN
         If (.Not.SetGen) THEN
           Call Warnin('Please set the general conditions first',
     &         .true.,12)
         ELSE
           Call Fluxes                                   {set the flux changes}
         ENDIF
       ELSE IF (Choice.EQ.5) THEN
         If (.Not.SetGen) THEN
           Call Warnin('Please set the general conditions first',
     &         .true.,12)
         ELSE
           Call Report                                   {specify when reports are to be generated}
         ENDIF
       ELSE IF (Choice.EQ.6) THEN
         Call StnSet                                     {set standard default run parameters}
       ELSE IF (Choice.EQ.7) THEN
         If (SetGen.AND.StSoil.AND.Nreps.GT.0.AND.Nspecs.GT.0.AND.
     &       Nflux.GT.0) THEN
           Call PoreVl                                   {suggest a conversion to pore volumes}
           Call Record                                   {record the parameter set}
         ELSE
           IF (.not.SetGen) THEN                         {but warn if the parameter set is incomplete}
             Call Warnin('You must set the general parameters before '
     &         //'saving the parameters',.true.,12)
           ELSE If (.Not.StSoil) THEN
             Call Warnin('You must set the soil parameters before '
     &         //'saving the parameters',.true.,12)
           ELSE If (Nspecs.LE.0) THEN
             Call Warnin('You must enter at least 1 boundary condit'
     &         //'ion set before saving the parameters',.true.,12)
           ELSE If (Nreps.LE.0) THEN
             Call Warnin('You must enter at least 1 report'
     &         //'time before saving the parameters',.true.,12)
           ELSE If (Nflux.LE.0) THEN
             Call Warnin('You must enter at least 1 flux condit'
     &         //'ion set before saving the parameters',.true.,12)
           ENDIF
         ENDIF
       ELSE IF (Choice.EQ.8) THEN
         Goto 10                                         {reset the program to a start-up condition}
       ELSE If (Choice.EQ.9) THEN
         If (SetGen.Or.StSoil.OR.Nspecs.GT.0.OR.Nflux.GT.0.OR.
     &       Nreps.GT.0) THEN
           Call Notice('Accessing a parameter will destroy the '
     &       //'information currently in memory.',.true.,10)
           If (.not.AskQus('Continue anyway?',.false.,12)) Goto 20
         ENDIF
         SetGen = .true.
         StSoil = .true.
         Call GetPar                                     {read a pre-existing parameter file}
       ENDIF
       If (Choice.NE.0) Goto 20
     END
     Subroutine SetUp
```

```
C-----------------------------------------------------------------------------------------------------------------------
C this routine sets and displays the standard simulation settings
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE 'SetHydra.CMB'
      Feedbk = .false.
      Icheck = 5
      TLimit = 0.5
      ItrLim = 20
      DtStep = 0.05
      EffZer = -30
      IonExB = .true.
      RevspB = .true.
      IrrevB = .true.
      MicroB = .true.
      AutoXB = .true.
      HydroB = .true.
      SelH2C = 1.0
      DO 10 I = 1,2
        Kinetc(I) = .true.
        DO 5 J = 1,4
          Rsorb(I,J) = 1.0
          Rsorb(I,4+J) = 1.0
    5     Rdgrad(I,J) = 1.0
        Do 10 J = 1,3
   10     RautoX(I,J) = 1.0
      Do 15 I = 1,4
   15   Scale(I) = 1.0
      Baseph = 7.8
      AmbnPh = 7.8
      Sponge = 0.0
      Return
      END
      Subroutine RunPrm(BinSet)
C-----------------------------------------------------------------------------------------------------------------------
C this routine sets and displays the general simulation settings
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE 'SetHydra.CMB'
      Dimension Query(9),Answer(9),String(9)
      Character Query*80,String*80
      Real Lo2Rel
      Logical Rel2Lo,BinSet,BadVal
      Data Nitems/9/,LenQry/50/,Query/
     & '0500CC ---General Simulation Parameters---',
     & '0701LD Modify/Examine specific Source/Sink settings',
     & '0902S? File to receive results',
     & '1103S? Title for run',
     & '1404R? Column length (cm)',
     & '1505R? Time to terminate simulation (sec)',
     & '1606R? Dx (distance between nodes) (cm)',
     & '1707R? Maximum Dt (time step) (sec)',
     & '1908LD Set BTC scale factors'/
     & Answer/2*0.0,15.0,80.0,5*0.0/
    1 Answer(2) = Lo2Rel(.false.)
      Answer(9) = Lo2Rel(.false.)
      If (.not.BinSet) THEN
        Do 10 I = 3,8
   10     Query(I)(6:6) = '?'
        BinSet = .true.
      ELSE
        Do 15 I = 2,8
   15     Query(I)(6:6) = 'D'
        String(3) = Fprint
        String(4) = Rlabel
        Answer(5) = Column
        Answer(6) = TmStop
        Answer(7) = Dx
        Answer(8) = DtMax
      ENDIF
      BadVal = .false.
      Call MenFil(LenQry,Nitems,Query,Answer,String,.true.)
      Fprint = String(3)
      Rlabel = String(4)
        Call StrLen(Rlabel,Length)
        If (Length.LE.0) Rlabel = 'Title not set'
      Column = Answer(5)
      If (Column.LE.0.0) THEN
        Call Warnin('Column must be > 0.0 cm long',.true.,12)
        Query(5)(6:6) = '?'
        BadVal = .true.
      ENDIF
      TmStop = Answer(6)
      If (TmStop.LE.0.0) THEN
        Call Warnin('Time to stop must be > 0.0 sec',.true.,12)
        Query(6)(6:6) = '?'
        BadVal = .true.
      ENDIF
      Dx = Answer(7)
      If (Dx.LE.0.0) THEN
        Call Warnin('Depth increment must be > 0.0 cm',.true.,12)
        Query(7)(6:6) = '?'
        BadVal = .true.
      ENDIF
      DtMax = Answer(8)
      If (DtMax.LE.0.0) THEN
```

```
        Call Warnin('Dt must be > 0.0 sec',.true.,12)
        Query(8)(6:6) = '?'
        BadVal = .true.
      ENDIF
      If (Rel2Lo(Answer(9))) Call ScaleF
      If (Rel2Lo(Answer(2))) Call Sinks
      If (BadVal.OR.Rel2Lo(Answer(2))) Goto 1
      Return
      END
      Subroutine ScaleF
C-----------------------------------------------------------------------------------------------------------------------
C this routine sets the BTC scale factors
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE 'SetHydra.CMB'
      Dimension Query(6),Answer(6),String(6)
      Character Query*80,String*80
      Real Lo2Rel
      Logical Rel2Lo,BadVal
      Data Nitems/6/,LenQry/60/,Query/
     & '0500CC ---BTC Scale Factors---',
     & '0600CC (BTC values = Absolute concentration/scale factor)',
     & '1001RD Scale Factor for ---(Hydrazine   (N2H4) )---',
     & '1102RD Scale Factor for ---(Hydrazinium (N2H5+))---',
     & '1203RD Scale Factor for ---(Calcium     (Ca++) )---',
     & '1304RD Scale Factor for ---(Oxygen      (O2)   )---'/
    1 Do 10 I = 3,6
   10   Answer(I) = Scale(I-2)
      BadVal = .false.
      Call MenFil(LenQry,Nitems,Query,Answer,String,.true.)
      DO 20 I = 3,6
        Scale(I-2) = Answer(I)
   20   If (Scale(I-3).LE.0.0) BadVal = .true.
      If (BadVal) Call Warnin(
     &       'Scale factors must be greater than zero',.true.,12)
      If (BadVal) Goto 1
      Return
      END
      Subroutine SoilVl(BinSet)
C-----------------------------------------------------------------------------------------------------------------------
C this routine sets and displays the soil characteristics
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE 'SetHydra.CMB'
      Dimension Query(5),Answer(5),String(5)
      Character Query*80,String*80
      Logical BadVal,AskQus,BinSet
      Data Nitems/5/,LenQry/60/,Query/
     & '0500CC ---Characteristics of the soil to be modeled---',
     & '0901R? Bulk density of the soil (gm/cc)',
     & '1102R? Dispersion (cm2/sec)',
     & '1303R? Saturated Water Content (cm3/cm3)',
     & '1504RD Hydrazinium permanent binding capacity (meq/gm)'/
    1 Answer(5) = Sponge
      If (.Not.BinSet) THEN
        Do 10 I = 1,Nitems-1
   10     If (Query(I)(5:5).NE.'C') Query(I)(6:6) = '?'
        BinSet = .true.
      ELSE
        Do 15 I = 1,Nitems
   15     If (Query(I)(5:5).NE.'C') Query(I)(6:6) = 'D'
        Answer(2) = BulkDn
        Answer(3) = Disper
        Answer(4) = ThetaS
      ENDIF
      BadVal = .false.
      Call MenFil(LenQry,Nitems,Query,Answer,String,.true.)
      BulkDn = Answer(2)
      If (BulkDn.LE.1.0.OR.BulkDn.GE.4.0) THEN
        Call Warnin('Bulk density must be > 1.0 & < 4.0',.true.,12)
        BadVal = .true.
        Query(2)(6:6) = '?'
      ENDIF
      Disper = Answer(3)
      If (Disper.LE.0.0) THEN
        Call Warnin('Dispersion must be greater than 0',.true.,12)
        Query(3)(6:6) = '?'
        BadVal = .true.
      ENDIF
      ThetaS = Answer(4)
      If (ThetaS.LE.0.0) THEN
        Call Warnin('Water Content must be greater than 0',.true.,12)
        Query(4)(6:6) = '?'
        BadVal = .true.
      ENDIF
      Sponge = ANswer(5)
      If (Sponge.LT.0.0) THEN
        Call Warnin('Hydrazinium binding capacity cannot be negative',
     &          .true.,12)
        Query(5)(6:6) = '?'
        BadVal = .true.
      ENDIF
      If (BadVal) Goto 1
      Return
      END
      Subroutine Bounds
```

```
C-------------------------------------------------------------------------------------------------------------------------------
C this routine sets and displays the boundary conditions
C-------------------------------------------------------------------------------------------------------------------------------
      INCLUDE 'SetHydra.CMB'
      Dimension Query(9),Answer(9),String(9)
      Character Query*80,String*80,NumStr*2
      Logical BadVal,AskQus
      Data Nitems/9/,LenQry/60/,Query/
     &'0500CC ---Description of the boundary (initial) conditions---',
     &'0701R? Depth in column at which conditions apply',
     &'1000CC ---Concentrations of Active Species---',
     &'1203R? Concentration of hydrazine in solution (meq/cc)',
     &'1304R? Concentration of hydrazinium in solution (meq/cc)',
     &'1405R? Concentration of sorbed hydrazinium (meq/g)',
     &'1506R? Concentration of Calcium in solution (meq/cc)',
     &'1607R? Concentration of sorbed Calcium (meq/g)',
     &'1708R? Concentration of oxygen in solution (meq/cc)'/
    1 If (Nspecs.Gt.0) THEN
        If (AskQus('Do you wish to review a set of previously entered'
     &    //' conditions? (Y/N):N'//char(8),.true.,12)) THEN
          Call In2Chr(Nspecs,NumStr)
          Call Notice('\Please enter which set is needed (1 to '//
     &      NumStr//'):',.false.,14)
          Read(*,*,Err=1) Ispec
          If (Ispec.LT.1.OR.Ispec.GT.Nspecs) Goto 1
        ELSE
          Ispec = Nspecs + 1
        ENDIF
      ELSE
        Ispec = 1
      ENDIF
    5 BadVal = .false.
      If (Ispec.GT.MaxSet) THEN
        Call Warnin('Maximum number of sets now in memory. No more '
     &    //'can be added.',.true.,12)
      ELSE
        If (Ispec.GT.Nspecs) THEN
        Do 10 I = 1,Nitems
   10     If (Query(I)(5:5).NE.'C') Query(I)(6:6) = '?'
        ELSE
          Do 15 I = 1,Nitems
   15       If (Query(I)(5:5).NE.'C') Query(I)(6:6) = 'D'
          Answer(2) = Depth(Ispec)
          Do 16 I = 1,6
   16       Answer(3+I) = Conc(Ispec,I)
        ENDIF
        Call MenFil(LenQry,Nitems,Query,Answer,String,.true.)
        Depth(Ispec) = Answer(2)
        If (Depth(Ispec).LT.0.OR.Depth(Ispec).GT.Column) THEN
          Call Warnin('Depth must be > 0 & less than column',.true.,12)
          Query(2)(6:6) = '?'
          BadVal = .true.
        ENDIF
        Do 20 I = 1,6
          Conc(Ispec,I) = Answer(3+I)
          If (Conc(Ispec,I).LT.0.0) THEN
            Call Warnin('Concentrations must be >= 0.0',.true.,12)
            Query(3+I)(6:6) = '?'
            BadVal = .true.
          ENDIF
   20   Continue
        If (BadVal) Goto 5
        If (Ispec.Gt.Nspecs) Nspecs = Nspecs + 1
      ENDIF
      Return
      END
      Subroutine Fluxes
C-------------------------------------------------------------------------------------------------------------------------------
C this routine sets and displays the input flux settings
C-------------------------------------------------------------------------------------------------------------------------------
      INCLUDE 'SetHydra.CMB'
      Dimension Query(7),Answer(7),String(7)
      Character Query*80,String*80,NumStr*2
      Real Lo2Rel
      Logical Rel2Lo,BadVal,AskQus
      Data Nitems/7/,LenQry/60/,Query/
     & '0500CC ---Description of material entering the column---',
     & '0701R? Time of the flux change',
     & '1202R? Imposed Flux (cm/hr) into column',
     & '1503R? Concentration of hydrazine entering (meq/cc)',
     & '1604R? Concentration of hydrazinium entering (meq/cc)',
     & '1705R? Concentration of Calcium entering (meq/cc)',
     & '1806R? Concentration of oxygen entering (meq/cc)'/
    1 If (Nflux.Gt.0) THEN
        If (AskQus('Do you wish to review a set of previously entered'
     &    //' conditions? (Y/N):N'//char(8),.true.,12)) THEN
          Call In2Chr(Nflux,NumStr)
          Call Notice('\Please enter which set is needed (1 to '//
     &      NumStr//'):',.false.,14)
          Read(*,*,Err=1) Ispec
          If (Ispec.LT.1.OR.Ispec.GT.Nflux) Goto 1
        ELSE
          Ispec = Nflux + 1
        ENDIF
```

```
      ELSE
        Ispec = 1
      ENDIF
    5 BadVal = .false.
      If (Ispec.GT.MaxSet) THEN
        Call Warnin('Maximum number of sets now in memory. No more '
     &    //'can be added.',.true.,12)
      ELSE
        If (Ispec.GT.Nflux) THEN
          Do 10 I = 1,Nitems
   10       If (Query(I)(5:5).EQ.'R') Query(I)(6:6) = '?'
          If (Ispec.GT.1) THEN
            Query(3)(6:6) = 'D'
            Answer(3) = FluxAm(1)
          ENDIF
        ELSE
          Do 15 I = 1,10
   15       If (Query(I)(5:5).EQ.'R') Query(I)(6:6) = 'D'
          Answer(2) = FluxTm(Ispec)
          Answer(3) = FluxAm(Ispec)
          Do 20 I = 1,4
   20       Answer(3+I) = FlConc(Ispec,I)
        ENDIF
        Call MenFil(LenQry,Nitems,Query,Answer,String,.true.)
        FluxTm(Ispec) = Answer(2)
        If (FluxTm(Ispec).LT.0.0.OR.FluxTm(Ispec).Gt.TmStop) THEN
          Call Warnin('Flux times must be > 0 & less than the '
     &          //'termination time',.true.,12)
          Query(2)(6:6) = '?'
          BadVal = .true.
        ENDIF
        FluxAm(Ispec) = Answer(3)
        If (FluxAm(Ispec).LT.0.0) THEN
          Call Warnin('Flux values must be >= 0.0',.true.,12)
          Query(3)(6:6) = '?'
          BadVal = .true.
        ELSE If (Ispec.GT.1.AND.FluxAm(Ispec).NE.FluxAm(1)) THEN
          Call Warnin('All flux values must be the same',.true.,12)
          Query(3)(6:6) = '?'
          BadVal = .true.
        ENDIF
        Do 25 I = 1,4
          FlConc(Ispec,I) = Answer(3+I)
          If (FlConc(Ispec,I).LT.0.0) THEN
            Call Warnin('Concentration must be >= 0.0',.true.,12)
            Query(3+I)(6:6) = '?'
            BadVal = .true.
          ENDIF
   25   Continue
        If (BadVal) Goto 5
        If (Ispec.GT.Nflux) Nflux = Nflux + 1
      ENDIF
      Return
      END
      Subroutine Report
C-----------------------------------------------------------------------------------------------------------------------
C this routine sets and displays the report times
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE 'SetHydra.CMB'
      Dimension Query(2),Answer(2),String(2)
      Character Query*80,String*80,NumStr*2
      Logical BadVal,AskQus
      Data Nitems/2/,LenQry/60/,Query/
     & '0500CC ~~~Specification of when reports are to be generated',
     & '1201R? Time (sec) of desired report'/
    1 If (Nreps.Gt.0) THEN
        If (AskQus('Do you wish to review a previously entered'
     &    //' time? (Y/N):N'//char(8),.true.,12)) THEN
          Call In2Chr(Nreps,NumStr)
          Call Notice('\Please enter which time is needed (1 to '//
     &      NumStr//'):',.false.,14)
          Read(*,*,Err=1) Ispec
          If (Ispec.LT.1.OR.Ispec.GT.Nreps) Goto 1
        ELSE
          Ispec = Nreps + 1
        ENDIF
      ELSE
        Ispec = 1
      ENDIF
    5 BadVal = .false.
      If (Ispec.GT.MaxSet) THEN
        Call Warnin('Maximum number of times now in memory. No more '
     &    //'can be added.',.true.,12)
      ELSE
        If (Ispec.GT.Nreps) THEN
          Do 10 I = 1,2
   10       If (Query(I)(5:5).NE.'C') Query(I)(6:6) = '?'
        ELSE
          Do 15 I = 1,2
   15       If (Query(I)(5:5).NE.'C') Query(I)(6:6) = 'D'
          Answer(2) = RepTim(Ispec)
        ENDIF
        Call MenFil(LenQry,Nitems,Query,Answer,String,.true.)
        RepTim(Ispec) = Answer(2)
```

```
        If (RepTim(Ispec).LT.0.0.OR.RepTim(Ispec).Gt.TmStop) THEN
          Call Warnin('report times must be > 0 & less than the '
     &          //'termination time',.true.,12)
          Query(2)(6:6) = '?'
          BadVal = .true.
        ENDIF
        If (BadVal) Goto 5
        If (Ispec.GT.Nreps) Nreps = Nreps + 1
      ENDIF
      Return
      END
      Subroutine StnSet
C-----------------------------------------------------------------------------------------------------------------------
C this routine sets and displays the standard simulation settings
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE 'SetHydra.CMB'
      Dimension Query(7),Answer(7),String(7)
      Character Query*80,String*80
      Real Lo2Rel
      Logical Rel2Lo,BadVal
      Data Nitems/7/,LenQry/60/,Query/
     & '0500CC ---Miscellaneous simulation parameters---',
     & '0801LD Provide debugging feedback',
     & '1002ID Periodic check for external abort command',
     & '1203RD Iterative Tolerance Limit  (expressed as % e.g. 0.5%)',
     & '1404ID Iterative Pass Maximum (e.g. 20)',
     & '1605RD Step Amount for Dt change  (in fractions, e.g. 0.05)',
     & '1806RD Effective Zero Value exponent (10**EZV)'/
    1 BadVal = .false.
      Answer(2) = Lo2Rel(Feedbk)
      Answer(3) = Icheck
      Answer(4) = TLimit
      Answer(5) = ItrLim
      Answer(6) = DtStep
      Answer(7) = EffZer
      Call MenFil(LenQry,Nitems,Query,Answer,String,.true.)
      Feedbk = Rel2Lo(Answer(2))
      Icheck = Answer(3)
      If (Icheck.LE.0) THEN
        Call Warnin('Abort command period must be > 0',.true.,12)
        Icheck = 5
        BadVal = .true.
      ENDIF
      TLimit = Answer(4)
      If (Tlimit.LE.0.0.OR.Tlimit.GT.10.0) THEN
        Call Warnin('Tolerance Limit must > 0% & <= 10.0%',.true.,12)
        Tlimit = 0.5
        BadVal = .true.
      ENDIF
      ItrLim = Answer(5)
      If (ItrLim.LE.0) THEN
        Call Warnin('Iteration limit must be > 0',.true.,12)
        ItrLim = 20
        BadVal = .true.
      ENDIF
      DtStep = Answer(6)
      If (DtStep.LE.0.0.OR.DtStep.GT.1.0) THEN
        Call Warnin('The Dt increment must be > 0.0 & < 1.0',.true.,12)
        DtStep = 0.05
        BadVal = .true.
      ENDIF
      EffZer = Answer(7)
      If (EffZer.GE.0.0) THEN
        Call Warnin('The effective zero coeff. must be < 0',.true.,12)
        EffZer = -30
        BadVal = .true.
      ENDIF
      If (BadVal) Goto 1
      Return
      END
      Subroutine Record
C-----------------------------------------------------------------------------------------------------------------------
C this routine sets and displays the standard simulation settings
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE 'SetHydra.CMB'
      Character Fsave*50,YesNo*3,NumStr*1
    1 Call GtFlNm('Parameters to be saved to...',Fsave)
        Open(Unit=12,FIle=Fsave,Status='NEW',Err=1)
      Write(12,1000) Fprint//' : Filename for results storage'
 1000   Format(A)
      Call StrLen(Rlabel,Length)
      If (length.LE.0) Length = 1
      Write(12,1000) Rlabel(1:Length)
 1010   Format(12X,A,' :',A)
 1020   Format(G15.8,' :',A)
      Write(12,1020) Column,'Column length (cm)'
      Write(12,1020) TmStop,'Time to terminate simulation (sec)'
      Write(12,1020) ThetaS,'Saturated Water Content'
      Write(12,1020) FluxAm(1)/3600.0,'Effective Conductivity (cm/sec)'
      Write(12,1020) BulkDn,'Bulk Density (gm/cc)'
      Write(12,1020) Disper,'Dispersion Coefficient (cm2/sec)'
      Write(12,1020) Sponge,'Hydrazinium binding capacity (meq/gm)'
      Write(12,1010) YesNo(IonExB),
     &  '---> Activate Ion-Exchange <---'
```

```
      If (IonExB) Write(12,1020) SelH2C,
     &                           'Selectivity of Hydrazinium to Calcium'
 1025   Format(A,' : ',A)
      Write(12,1010) YesNo(RevspB),
     &  '---> Activate Reversible Sorption <---'
       If (RevspB) THEN
         Write(12,1010) YesNo(Kinetc(1)),'Activate Kinetic Sorption'
         Write(12,1020) Rsorb(1,1),'Hydrazine Rate Coefficient (k1)'
         Write(12,1020) Rsorb(1,2),'Hydrazine Rate Coefficient (k2)'
         Write(12,1020) Rsorb(1,3),'Hydrazine exponent Coefficient (N)'
         Write(12,1020) Rsorb(1,4),'Hydrazine Rate Coefficient (kff)'
         Write(12,1020) Rsorb(1,5),'Hydrazine Rate Coefficient (kbb)'
         Write(12,1020) Rsorb(1,6),'Hydrazine Rate Coefficient (kp)'
         Write(12,1020) Rsorb(1,7),'Hydrazine Rate Coefficient (kq)'
         Write(12,1010) YesNo(Kinetc(2)),'Activate Kinetic Sorption'
         Write(12,1020) Rsorb(2,1),'Hydrazinium Rate Coefficient (k1)'
         Write(12,1020) Rsorb(2,2),'Hydrazinium Rate Coefficient (k2)'
         Write(12,1020) Rsorb(2,3),'Hydrazinium exponent Coefficient(N)'
         Write(12,1020) Rsorb(2,4),'Hydrazinium Rate Coefficient (kff)'
         Write(12,1020) Rsorb(2,5),'Hydrazinium Rate Coefficient (kbb)'
         Write(12,1020) Rsorb(2,6),'Hydrazinium Rate Coefficient (kp)'
         Write(12,1020) Rsorb(2,7),'Hydrazinium Rate Coefficient (kq)'
       ENDIF
       Write(12,1010)YesNo(IrrevB),
      &  '---> Activate Irreversible Sorption <---'
       If (IrrevB) THEN
         Write(12,1020) Rsorb(1,8),'Hydrazine Rate Coefficient (k3)'
         Write(12,1020) Rsorb(2,8),'Hydrazinium Rate Coefficient (k3)'
       ENDIF
       Write(12,1010) YesNo(MicroB),
      &  '---> Activate Microbial Degradation <---'
       If (MicroB) THEN
         Write(12,1020) Rdgrad(1,1),'Hydrazine Rate Coefficient (ka)'
         Write(12,1020) Rdgrad(1,2),'Hydrazine Rate Coefficient (kb)'
         Write(12,1020) Rdgrad(1,3),'Hydrazine Rate Coefficient (kc)'
         Write(12,1020) Rdgrad(1,4),'Hydrazine enzyme level (Eo)'
         Write(12,1020) Rdgrad(2,1),'Hydrazinium Rate Coefficient (ka)'
         Write(12,1020) Rdgrad(2,2),'Hydrazinium Rate Coefficient (kb)'
         Write(12,1020) Rdgrad(2,3),'Hydrazinium Rate Coefficient (kc)'
         Write(12,1020) Rdgrad(2,4),'Hydrazinium enzyme level (Eo)'
       ENDIF
       Write(12,1010) YesNo(AutoXB),
      &  '---> Activate Auto-Oxidation <---'
       If (AutoXB) THEN
         Write(12,1020) Rautox(1,1),'Hydrazine Rate Coefficient (ki)'
         Write(12,1020) Rautox(1,2),'Hydrazine Rate Coefficient (kii)'
         Write(12,1020) Rautox(1,3),'Hydrazine Rate Coefficient (kiii)'
         Write(12,1020) Rautox(2,1),'Hydrazinium Rate Coefficient (ki)'
         Write(12,1020) Rautox(2,2),'Hydrazinium Rate Coefficient (kii)'
         Write(12,1020)Rautox(2,3),'Hydrazinium Rate Coefficient (kiii)'
       ENDIF
       Write(12,1010) YesNo(HydroB),
      &  '---> Activate Hydrolysis Conversion <---'
       If (HydroB) THEN
         Write(12,1020) BasepH,'Base pH for conversion'
         Write(12,1020) AmbnpH,'Ambient pH'
       ENDIF
       Write(12,1040) Nspecs,'Number of boundary condition specs'
 1040   Format(I15,' :',A)
       Do 10 I = 1,Nspecs
         CEC = Conc(I,3) + Conc(I,5)
         Write(12,1020) Depth(I),'----> Depth (cm) <----'
         Write(12,1020) Conc(I,1),'Hydrazine in solution (meq/cc)'
         Write(12,1020) Conc(I,2),'Hydrazinium in solution (meq/cc)'
         Write(12,1020) Conc(I,3),'Hydrazinium sorbed (meq/g)'
         Write(12,1020) Conc(I,4),'Calcium in solution (meq/cc)'
         Write(12,1020) Conc(I,5),'Calcium sorbed (meq/g)'
         Write(12,1020) Conc(I,6),'Oxygen in solution (meq/cc)'
   10    Write(12,1020) CEC,'Cation Exchange Capacity (meq/g)'
       Write(12,1020) Dx,'Dx = Nodal increment (cm)'
       Write(12,1020) Dtmax,'Maximum Dt (secs)'
       Write(12,1040) Nflux,'Number of flux changes during run'
       Do 20 I = 1,Nflux
         Write(12,1020) FluxTm(I),'Time (sec) of flux change'
         Write(12,1020) FluxAm(I),'Flux (cm/hr) imposed'
         Write(12,1020) FlConc(I,1),'Hydrazine in solution (meq/cc)'
         Write(12,1020) FlConc(I,2),'Hydrazinium in solution (meq/cc)'
         Write(12,1020) FlConc(I,3),'Calcium in solution (meq/cc)'
   20    Write(12,1020) FlConc(I,4),'Oxygen in solution (meq/cc)'
       Write(12,1040) Nreps,'Number of reports to be generated'
       Do 30 I = 1,Nreps
   30    Write(12,1020) RepTim(I),'Report to be generated (sec)'
       Write(12,1020) Scale(1),'BTC Divisor scale factor for N2H4'
       Write(12,1020) Scale(2),'BTC Divisor scale factor for N2H5+'
       Write(12,1020) Scale(3),'BTC Divisor scale factor for Ca++'
       Write(12,1020) Scale(4),'BTC Divisor scale factor for O2'
       Write(12,1010) YesNo(FeedBk),'Feedback for debugging'
       Write(12,1040) Icheck,'abortion check periodicity'
       Write(12,1020) Tlimit,'Tolerance Limit in percentage'
       Write(12,1040)ItrLim,'Number of passes before iterative failure'
       Write(12,1020) DtStep,'Increment for Dt - fractional'
       Write(12,1020) EffZer,'Power for approximately zero value'
       Close(12)
       Return
```

```
      END
      Subroutine GetPar
C-----------------------------------------------------------------------------------------------------------------------
C this routine sets and displays the standard simulation settings
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE 'SetHydra.CMB'
      Logical Str2Lo
      Character Fsave*50,YesNo*3,NumStr*1,Aline*80
    1 Call GtFlNm('Parameters to be read',Fsave)
        Open(Unit=12,FIle=Fsave,Status='OLD',Err=1)
      Read(12,1000) Aline
      Call StrBeg(Aline,Istart)
      Call StrLen(Aline,Iend)
      Do 2 I = Istart,Iend
        If (Aline(I:I).EQ.' ') Goto 3
    2 Continue
    3 Fprint = Aline(Istart:I)
 1000   Format(A)
      Read(12,1000) Rlabel
 1010   Format(12X,A)
      Read(12,*) Column
      Read(12,*) TmStop
      Read(12,*) ThetaS
      Read(12,*) ConSat
      Read(12,*) BulkDn
      Read(12,*) Disper
      Read(12,*) Sponge
      Read(12,1010) YesNo
        IonExB = Str2Lo(YesNo)
        If (IonExB) Read(12,*) SelH2C
      Read(12,1010) YesNo
        RevspB = Str2Lo(YesNo)
        If (RevspB) THEN
          Do 10 I = 1,2
            Read(12,1010) YesNo
            Kinetc(I) = Str2Lo(YesNo)
            Do 10 J = 1,7
   10         Read(12,*) Rsorb(I,J)
        ENDIF
      Read(12,1010) YesNo
        IrrevB = Str2Lo(YesNo)
        If (IrrevB) THEN
          Read(12,*) Rsorb(1,8)
          Read(12,*) Rsorb(2,8)
        ENDIF
      Read(12,1010) YesNo
        MicroB = Str2Lo(YesNo)
        If (MicroB) THEN
          Do 20 I = 1,2
            Do 20 J = 1,4
   20         Read(12,*) Rdgrad(I,J)
        ENDIF
      Read(12,1010) YesNo
        AutoXB = Str2Lo(YesNo)
        If (AutoXB) THEN
          Do 30 I = 1,2
            Do 30 J = 1,3
   30         Read(12,*) RautoX(I,J)
        ENDIF
      Read(12,1010) YesNo
        HydroB = Str2Lo(YesNo)
        If (HydroB) THEN
          Read(12,*) BasepH
          Read(12,*) AmbnpH
        ENDIF
      Read(12,*) Nspecs
      Do 40 I = 1,Nspecs
        Read(12,*) Depth(I)
        Do 35 J = 1,6
   35     Read(12,*) Conc(I,J)
   40     Read(12,*) CEC
      Read(12,*) Dx
      Read(12,*) Dtmax
      Read(12,*) Nflux
      Do 50 I = 1,Nflux
        Read(12,*) FluxTm(I)
        Read(12,*) FluxAm(I)
        Do 50 J = 1,4
   50     Read(12,*) FlConc(I,J)
      Read(12,*) Nreps
      Do 60 I = 1,Nreps
   60   Read(12,*) RepTim(I)
      Do 65 I = 1,4
   65   Read(12,*) Scale(I)
      Read(12,1010) YesNo
        FeedBk = Str2Lo(YesNo)
      Read(12,*) Icheck
      Read(12,*) Tlimit
      Read(12,*) ItrLim
      Read(12,*) DtStep
      Read(12,*) EffZer
   70 Close(12)
      Return
      END
```

```
      Subroutine Sinks
C-----------------------------------------------------------------------------------------------------------------------
C this routine allows the examination and/or alteration of standard source/sink terms
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE 'SetHydra.CMB'
      Dimension Option(10)
      Character Option*80
      Integer Choice
      Data LenOpt/60/,Nitems/10/,Option/
     & 'Source/Sink Effects on Hydrazin(e)(ium)',' ',
     & '<0> Return to General Simulation Parameters Menu',' ',
     & '<1> Ion-Exchange (Hydrazium vs Calcium)',
     & '<2> Reversible Sorption',
     & '<3> Irreversible Sorption',
     & '<4> Microbial Degradation',
     & '<5> Auto-Oxidation',
     & '<6> Hydrolysis (Hydrazine <-> Hydrazium)'/
    1 Choice = MenChc(LenOpt,Nitems,Option,.true.)
      If (Choice.NE.0) THEN
        If (Choice.Eq.1) THEN
          Call IonExc
        ELSE If (Choice.EQ.2) THEN
          Call RevSrp
        ELSE If (Choice.EQ.3) THEN
          Call IrrevS
        ELSE If (Choice.EQ.4) THEN
          Call MicrDg
        ELSE If (Choice.EQ.5) THEN
          Call AutoOx
        ELSE If (Choice.EQ.6) THEN
          Call Hydrol
        ENDIF
        Goto 1
      ENDIF
      Return
      END
      Subroutine IonExc
C-----------------------------------------------------------------------------------------------------------------------
C this routine allows the examination and/or alteration of
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE 'SetHydra.CMB'
      Dimension Query(3),Answer(3),String(3)
      Character Query*80,String*80
      Real Lo2Rel
      Logical Rel2Lo,BadVal
      Data Nitems/3/,LenQry/60/,Query/
     & '0500CC ---- Factors controlling Ion-Exchange ---' ,
     & '0701LD Activate this source/sink effect',
     & '1202RD Selectivity Coefficient for Hydrazium->Calcium'/
    1 Answer(2) = Lo2Rel(IonExB)
      Answer(3) = SelH2C
      Badval = .false.
      Call MenFil(LenQry,Nitems,Query,ANswer,String,.true.)
      IonExB = Rel2Lo(Answer(2))
      SelH2C = Answer(3)
      If (IonExB.AND.SelH2C.LE.0.0) THEN
        Call Warnin('Selectivity must be greater than zero',.true.,12)
        SelH2C = 1.0
        BadVal = .true.
      ENDIF
      If (BadVal) Goto 1
      Return
      END
      Subroutine RevSrp
C-----------------------------------------------------------------------------------------------------------------------
C this routine allows the examination and/or alteration of  Reversible Sorption
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE 'SetHydra.CMB'
      Dimension Query1(6),Answr1(6),Strng1(6)
      Dimension Query2(10),Answr2(10),Strng2(10)
      Character Query1*80,Strng1*80,Query2*80,Strng2*80,Hspp(2)*15
      Real Lo2Rel
      Logical Rel2Lo,BadVal,Setion(2)
      Data Nitem1/6/,LenQry/60/,Query1/
     & '0500CC ---- Factors controlling Reversible Sorption ---',
     & '0701LD Activate this source/sink effect',
     & '0900CC ---Reversible sorption for Hydrazine---',
     & '1102LD Alter or set the rate coefficients',
     & '1500CC ---Reversible sorption for Hydrazinium---',
     & '1703LD Alter or set the rate coefficients'/
      Data Nitem2/10/,Query2/
     & '0500CC ---- Factors controlling Reversible Sorption for---',
     & '0700CC ',
     & '0901LD Use kinetic approach for reversible sorption',
     & '1002RD Forward reaction rate coefficient (k1)',
     & '1103RD Backwards reaction rate coefficient (k2)',
     & '1204RD Dissolved concentration exponential coefficient (N)',
     & '1305RD Forward reaction rate coefficient (kff)',
     & '1406RD Backwards reaction rate coefficient (kbb)',
     & '1507RD Binding reaction rate (kp)-from tightly sorbed',
     & '1608RD Binding reaction rate (kq)-from loosely sorbed'/
      Data Hspp/'Hydrazine','Hydrazinium'/
    1 Answr1(2) = Lo2Rel(RevspB)
      Answr1(4) = Lo2Rel(.false.)
```

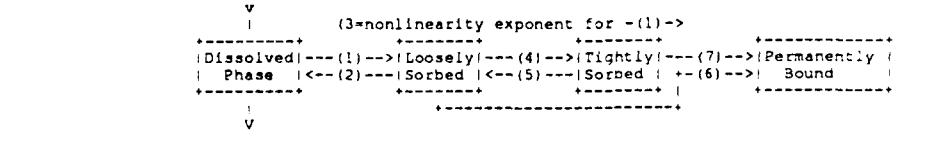
```
      Answr1(6) = Lo2Rel(.false.)
      Call MenFil(LenQry,Niteml,Queryl,Answrl,Strngl,.true.)
      RevSpB = Rel2Lo(Answrl(2))
      If (RevSpb) THEN
        SetIon(1) = Rel2Lo(Answrl(4))
        SetIon(2) = Rel2Lo(Answrl(6))
        Do 20 Ion = 1,2
          If (SetIon(Ion)) THEN
            Query2(2)(8:) = Hspp(Ion)
            Answr2(3) = Lo2Rel(Kinetc(Ion))
            Do 10 I = 1,7
   10         Answr2(3+I) = Rsorb(Ion,I)
            Call MenFil(LenQry,Nitem2,Query2,ANswr2,Strng2,.true.)
            Kinetc(Ion) = Rel2Lo(Answr2(3))
            Do 15 I = 1,7
   15         Rsorb(Ion,I) = Answr2(3+I)
          ENDIF
   20   Continue
        If (SetIon(1).OR.SetIon(2)) Goto 1
      ENDIF
      Return
      END
      Subroutine IrrevS
C-----------------------------------------------------------------------------------------------------------------------
C this routine allows the examination and/or alteration of irreversible sorption
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE 'SetHydra.CMB'
      Dimension Query(6),Answer(6),String(6)
      Character Query*80,String*80
      Real Lo2Rel
      Logical Rel2Lo
      Data Nitems/6/,LenQry/60/,Query/
     & '0500CC ---- Factors controlling Irreversible Sorption ---' ,
     & '0701LD Activate this source/sink effect',
     & '0900CC ---Irreversible sorption for Hydrazine---',
     & '1102RD Forward reaction rate coefficient (k3)',
     & '1400CC ---Irreversible sorption for Hydrazinium---',
     & '1603RD Forward reaction rate coefficient (k3)'/
      1 Answer(2) = Lo2Rel(IrrevB)
      Answer(4) = Rsorb(1,8)
      Answer(6) = Rsorb(2,8)
      Call MenFil(LenQry,Nitems,Query,ANswer,String,.true.)
      IrrevB = Rel2Lo(Answer(2))
      Rsorb(1,8) = Answer(4)
      Rsorb(2,8) = Answer(6)
      Return
      END
      Subroutine MicrDg
C-----------------------------------------------------------------------------------------------------------------------
C this routine allows the examination and/or alteration of
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE 'SetHydra.CMB'
      Dimension Query(12),Answer(12),String(12)
      Character Query*80,String*80
      Real Lo2Rel
      Logical Rel2Lo,BadVal
      Data Nitems/12/,LenQry/60/,Query/
     & '0500CC ---- Factors controlling Microbial Degradation ---',
     & '0701LD Activate this source/sink effect',
     & '0900CC ---Microbial Degradation for Hydrazine---',
     & '1002RD Forward rate coefficient to complex (ka)',
     & '1103RD Backwards rate coefficent from complex (kb)',
     & '1204RD Forward rate coeffcient to product (kc)',
     & '1305RD Size of enzyme complex (Eo)',
     & '1500CC ---Microbial Degradation for Hydrazinium---',
     & '1606RD Forward rate coefficient to complex (ka)',
     & '1707RD Backwards rate coefficent from complex (kb)',
     & '1808RD Forward rate coeffcient to product (kc)',
     & '1909RD Size of enzyme complex (Eo)'/
      1 Answer(2) = Lo2Rel(MicroB)
      Do 10 I = 1,4
        Answer(3+I) = Rdgrad(1,I)
   10   Answer(8+I) = Rdgrad(2,I)
      Call MenFil(LenQry,Nitems,Query,ANswer,String,.true.)
      MicroB = Rel2Lo(Answer(2))
      Do 20 I = 1,4
        Rdgrad(1,I) = Answer(3+I)
   20   Rdgrad(2,I) = Answer(8+I)
      Return
      END
      Subroutine AutoOx
C-----------------------------------------------------------------------------------------------------------------------
C this routine allows the examination and/or alteration of auto-oxidation
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE 'SetHydra.CMB'
      Dimension Query(10),Answer(10),String(10)
      Character Query*80,String*80
      Real Lo2Rel
      Logical Rel2Lo,BadVal
      Data Nitems/10/,LenQry/60/,Query/
     & '0500CC ---- Factors controlling Auto-Oxidation ---',
     & '0701LD Activate this source/sink effect',
     & '0900CC ---Auto-Oxidation of Hydrazine---',
     & '1002RD Forward rate coefficient to complex (k1)',
```

```
     & '1103RD Backwards rate coefficent from complex (kii)',
     & '1204RD Forward rate coeffcient to product (kiii)',
     & '1500CC ---Auto-Oxidation of Hydrazinium---',
     & '1605RD Forward rate coefficient to complex (ki)',
     & '1706RD Backwards rate coefficent from complex (kii)',
     & '1807RD Forward rate coeffcient to product (kiii)'/
    1 Answer(2) = Lo2Rel(AutoXB)
      Do 10 I = 1,3
        Answer(3+I) = RautoX(1,I)
   10   Answer(7+I) = RautoX(2,I)
      Call MenFil(LenQry,Nitems,Query,ANswer,String,.true.)
      AutoXB = Rel2Lo(Answer(2))
      Do 20 I = 1,3
        RautoX(1,I) = Answer(3+I)
   20   RautoX(2,I) = Answer(7+I)
      Return
      END
      Subroutine Hydrol
C-----------------------------------------------------------------------------------------------------------------------
C this routine allows the examination and/or alteration of
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE 'SetHydra.CMB'
      Dimension Query(4),Answer(4),String(4)
      Character Query*80,String*80
      Real Lo2Rel
      Logical Rel2Lo,BadVal
      Data Nitems/4/,LenQry/60/,Query/
     & '0500CC ---- Factors controlling Hydrolysis ---',
     & '0701LD Activate this source/sink effect',
     & '0902RD Critical base pH value',
     & '1103RD Ambient pH'/
    1 Answer(2) = Lo2Rel(HydroB)
      ANswer(3) = BasePh
      Answer(4) = AmbnPh
      Badval = .false.
      Call MenFil(LenQry,Nitems,Query,ANswer,String,.true.)
      BasePh = ANswer(3)
      AmbnpH = Answer(4)
      If (BasepH.LT.1.0.OR.BasepH.GT.14.0.OR.
     &    AmbnpH.LT.1.0.OR.AmbnpH.GT.14.0)   THEN
        Call Warnin('pH must be between 1.0 and 14.0',.true.,12)
        BadVal = .true.
      ENDIF
      If (BadVal) Goto 1
      Return
      END
      Subroutine PoreVl
C-----------------------------------------------------------------------------------------------------------------------
C this routine allows the conversion into pore volumes
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE 'SetHydra.CMB'
      Logical AskQus
      PVlsec = ThetaS*Column*3600.0/FluxAm(1)
      Call Notice('You may have entered times in pore volumes',
     &            .true.,5)
      Call Vtab(8)
      Write(*,1000) ('Flux changes at:',FluxTm(I),FluxTm(I)*PvlSec,
     &                                  I=1,Nflux)
      Write(*,1000) ('Reports at     :',RepTim(I),RepTim(I)*PvlSec,
     &                                  I=1,Nreps)
      Write(*,1000)  'Termination at :',TmStop,TmStop*PVLSec
 1000   Format(2X,A,G12.5,' pvl--> ',G12.5,' sec')
      If (AskQus('Do you wish to make these conversions? (Y/N): Y'
     &       //char(8),.false.,20)) THEN
        Do 10 I = 1,Nflux
   10     FluxTm(I) = FluxTm(I)*PVLSec
        Do 20 I = 1,Nreps
   20     RepTim(I) = RepTim(I)*PVlSec
        TmStop =TmStop*PVLSec
      ENDIF
      Return
      END
------------NOTE: See UTILITY SUBROUTINES for any routines used here that are not included here--------
```

B. Program for simulation of hydrazin(e)(ium) movement in a saturated column

HYDRAZIN Variable Dictionary

```
Adsorb(2,MaxNod)        = amount of material absorbed at each node (1=old, 2=now)
AmbnPh(MaxNod)          = ambient pH at each node - usedin hydrolysis calculations
Apprx0                  = approximately zero - the effective floor which should be slightly non-zero
AutoOx                  = boolean for auto-oxidation (true -> auto-oxidation does occur)
BalSol                  = total solute in the system - used in mass balance calculations
BalSor                  = total sorbed material in the system - used in mass balance calculations
BasePh                  = base pH value used in hydrolysis calculations
BdOpen(2,MaxNod)        = booleans indicating whether there is any remaining capacity to bind at each node
Bound(2,MaxNod)         = the amount of hydrazinium bound permanently at each node
Bulkdn                  = bulk density (gm/cc)
Cact(MaxIon)            = the known (inflow + initial contents) amount of each ion in the system
CDissv(MaxIon,MaxNod)   = the solute concentration (meq/cc) for each ion at each node
Cecfix                  = the total cation exchange capacity at each node
Cinit(MaxIon)           = the initial amount of each ion in the system at the start of the simulation
ColDep(MaxSet)          = the depth corresponding to each boundary condition specification
Colion(MaxSet,MaxIon,2)= the concentration (meq/cc) of each ion at each boundary condition specification
Column                  = the length of the simulated column in cm
ConSat                  = saturated conductivity (cm/sec)
Cpred(MaxIon,0:1)       = the predicted (outflow + current contents) amount of each ion in the system
Csorb(MaxIon,MaxNod)    = the sorbed concetration (meq/gm) of each ion at each node
Diagon(MaxNod)          = the vector holding the diagonal elements of the tridiagonal solution system
Disper                  = dispersion coefficient (cm**2/sec)
Dt                      = the current time increment (delta T) (sec)
DtAltr                  = boolean indicating the need to change the Dt (if true)
DtDecr                  = the amount that Dt will be decremented
DtDown                  = boolean indicating the direction of Dt change
DtLast                  = the value of Dt before the change
Dtmax                   = the maximum value that Dt is allowed to assume
Dtmin                   = the minimum value that Dt is allowed to assume
DtSpan                  = the time needed to change Dt from Dtmin to DtMax
DtStar                  = boolean indicating that Dt is to be changed
DtStep                  = the fraction of Dtmax to be added to Dt if the change is ascending
Dtsum                   = the sum of all Dt values throughout the simulation (used in calculating mean Dt)
Dx                      = the distance betwen nodes (cm)
Errors(0:MaxIon,0:1)    = the percent error for each ion
Extrem(2,MaxIon,0:4)    = a set of min/max/sum/sum-of-squares values for reporting extremes of certain variables
Fbrkth                  = file name for the breakthrough values (15 characters)
Feedbk                  = boolean used to turn on (.true.) or off (.false.) debugging statements
FixSrb(2,MaxNod)        = the concentration (meq/gm) of tighly bounded absorbed material
Flowin(0:MaxIon)        = the cummulative amount (meq) of each ion that has entered the system
Flowou(0:MaxIon)        = the cummulative amount (meq) of each ion that has left the system
Fluxam(MaxSet,0:MaxIon)= the input flux (0) or concentrations (1..MaxIon) for each flux/concetration change
Fluxtm(MaxSet)          = the time (sec) at which each flux/concentration change time occurs
Fprint                  = the filename for the report (results) information (15 characters)
GoIonX                  = boolean indicating that ion exchange is to occur (when .true.)
HydrIN(MaxIon)          = the amount of hydrazin(e)(ium) converted to the other form upon entry into the system
HydrLa(MaxIon)          = the total amount of hydrolysis conversion that has occurred
Hydrol                  = boolean indicating that hydrolysis is to occur (when .true.)
Icheck                  = indicates how frequently the abort file should be checked
IflxPt                  = a pointer that indicates which of the flux changes is currently active
Ionlst                  = indicates which of all possible ions in the system is in fact the 1st active ion
Irepet                  = the counter of the number of completed time steps
Irrevs                  = boolean indicating whether irreversible sorption is to take place
Itmptr                  = pointer indicating which of the report times is the next to be active
ItrLim                  = the iterative limit - the number of times the system will attempt to find a converging solution
ItrTrk(30)              = stores the cummulative number of iterations needed to acheive a convergent solution
Kinetc(2)               = booleans indicating that a kinetic (as opposed to an equilibrium) approach is to be used (N2H4 &
N2H5+)
LckDwn                  = counter of the number of iterations after a flux change (if < 10, then Dt stays at Dtmin)
MaxIon                  = maximum number of ion in the system (=4 : N2H4, N2H5+, Ca++, H+ in that order)
MaxNod                  = maximum number of nodes in the system (=202)
MaxRep                  = maximum number of reports allowed (=10)
MaxSet                  = maximum number of boundary condition specification sets allowed (=7)
Microb                  = boolean indicating whether microbial degradation is to be used
NdtSum                  = accumulates the number of Dt changes and is used to calculate the mean Dt
Nflux                   = number of flux changes to occur during the simulation
NinUse                  = number of ions in use
Nions                   = total number of ions in the system (=4)
Nodes                   = the number of nodes in the system
Nrepr                   = the number of reports requested during the simulation
Nspecs                  = the number of boundary condition sets used to setup the system initially
OldAds(2,MaxNod)        = the old (previous) adsorbed amounts for the 2 hydrazine species at each node
OldBnd(2,MaxNod)        = the old (previous) permanently bounded amounts for N2H4 and N2H5+ at each node
OldDis(MaxIon,MaxNod)   = the old (previous) dissolved material for each ion at each node
OldFix(2,MaxNod)        = the old (previous) tighty bound material for N2H4 and N2H5+ at each node
OldFlx                  = the old (previous) flux into the system
OldSor(MaxIon,MaxNod)   = the old (previous) material sorbed (in the CEC) for each ion at each node
OxNitr(MaxNod)          = the oxygen/hydrazin complex used in autooxidation calculations
PoreVl                  = the cummulative amount of water that has left the system (pore volumes)
RatIrv(2)               = rate coefficients for irreversible sorption (N2H4 & N2H5+ - 1st order only)
RatRev(2,7)             = rate coefficients for reversible sorption (N2H4 & N2H5+, where:
```




```
RautoX(2,3)             = auto-oxidation coefficents
Requil(2)               = equilibrium coefficents (replacing -(1)-> & <-(2)- for N2H4 & N2H5+)
Result(MaxNod)          = result vector used in the tridiagonal solution
Revers                  = boolean indicating that reversible sorption is to be used
Rlabel                  = 80 character string labeling the particular simulation
```

```
Rmicrb(2,4)          = rate coefficents for microbial degradation (for N2H4 & N2H5+ with only (.,1) active
SbDiag(MaxNod)       = vector holding the subdiagonal coefficients for use in the tridiagonal solution
SCa2Hy               = selectivity value between Ca++ and N2H5+
Scale(MaxIon)        = divisors of output concentrations
Solute(0:MaxIon)     = text strings for the various active entries (water,N2H4,N2H5+,Ca++,H+)
SpDiag(MaxNod)       = vector holding the super-diagonal coefficents for the tridiagonal solution
Sponge               = maximum capacity to permanently bind N2H5+ at each node
SrbOut(MaxIon)       = total amount of material lost to the system due to sorption
ThetaS               = saturated water content
Time                 = value of the simulation clock (sec)
Tlevel               = tolerance level (in percent)-solution is accepted when max diff at any node between
                       sucessive solutions drop down to or below this value
TmLast               = last time at which a breakthrough point was generated
TmRepr(0:MaxRep)     = times at which reports are to be generated
TmSpan               = time span between breakthrough points
Tmstop               = time to stop the simulation
TotNow(MaxIon,5)     = holds the totals (meq) in the various simulated compartments
UseIon(MaxIon)       = booleans which indicate whether a given ion is active
Valenc(MaxIon)       = valences of the various ions
Veloc                = pore water velocity
Winit                = initial water total
```

```
      Implicit Double Precision (A-H,O-Z)
      SAVE
      Parameter(MaxIon=4, MaxNod=202, MaxSet=7, MaxRep=10)
      Common/Contrl/ApprxO,CecFix,Column,Dt,Dtmax,Dtmin,DtStep,Dx,
     &              Icheck,Ionist,Irepet,Itmptr,ItrLim,NinUse,Nions,
     &              Nodes,Nrepr,PoreVl,Scale(MaxIon),ThetaS,Time,
     &              Tlevel,TmLast,TmRepr(0:MaxRep),Tmstop,Winit,
     &              DtDecr,DtLast,DtSpan,OldFlx,LckDwn,Sponge
      Common/Errors/BalSol,BalSor,Cact(MaxIon),Cpred(MaxIon,0:1),
     &              Dtsum,Errors(0:MaxIon,0:1),Extrem(2,MaxIon,0:4),
     &              ItrDrx(30),NdtSum,TotNow(MaxIon,5)
      Common/Flux   /FluxIm(MaxSet),FluxAm(MaxSet,0:MaxIon),NFlxAt,IFlxPt,
     &              TmSpan,Veloc
      Common/Ions   /CDissv(MaxIon,MaxNod),OldDis(MaxIon,MaxNod),
     &              Csorb(MaxIon,MaxNod),OldSor(MaxIon,MaxNod),
     &              Adsorb(2,MaxNod),OldAds(2,MaxNod),Valenc(MaxIon),
     &              FlxSrb(2,MaxNod),OldFlx(2,MaxNod),Bound(2,MaxNod),
     &              OldBnd(2,MaxNod),BdOpen(2,MaxNod)
      Common/Logics/AutoDx,DtAltr,DtDown,DtStar,Feedbk,GoIonX,Hydrol,
     &              Irrevs,Kinetc(2),Microb,Revers,UseIon(MaxIon)
      Common/Manual/Qin(MaxIon),FlowIn(0:MaxIon),FlowOu(0:MaxIon),
     &              HydrIN(MaxIon),HydrIa(MaxIon),GrsOut(MaxIon)
      Common/SetUp  /CoIon(MaxSet,MaxIon,2),CoIDep(MaxSet),NSpecs
      Common/Soil   /BulkDn,ConSat,Disper,GCa2Hy
      Common/Rates  /AmbrPh(MaxNod),BaseMn,OxNitr(MaxNod),RatIrv(2),
     &              RatRev(2,7),RautoX(2,3),Requil(2),Rmicrb(2,4)
      Common/TriDat/SbDiag(MaxNod),Diagon(MaxNod),SpDiag(MaxNod),
     &              Result(MaxNod)
      Common/Comchr/Fbrkth,Fprint,Rlabel,Solute(0:MaxIon)
      Character Rlabel*80,Solute*16,Fprint*15,FbrkTh*15
      Logical AutoDx,DtAltr,DtDown,DtStar,Feedbk,GoIonX,Hydrol,Irrevs,
     &        Kinetc,Microb,Revers,UseIon,BdOpen
```

FORTRAN 77 code for HYDRAZIN

```
      PROGRAM Hydraz
C-----------------------------------------------------------------------------------------------------------------
C                  Dr. Stephen A. Bloom - for Dr. Robert Manase., Soil Science 01/16/88
C-----------------------------------------------------------------------------------------------------------------
      INCLUDE 'Hydrazin.CMB'
      Call StartP                                                    !read the simulation run parameters & initia.ize
C-------REPEAT-------------------------------------
   10 Call DtChan                                                    !check whether Dt needs changing;
      TIME = TIME + DT                                               !increment the clock;
      IREPET = IREPET + 1                                            !increment the iteration counter;
      PoreVl = FlowOu(0) / (Column*ThetaS)                           !set the pore volume)
      Call ChkFix                                                    !check for flux changes)
      IF (Time.GE.TMREPR(Itmptr)) THEN                               !check if a report is needed)
        Call RepIon                                                  !generate an error & profile report;
        Call RepPrf                                                  !generate an extract of the profile;
      ENDIF
      If (PoreVl.GT.1.0E-06) THEN                                    !if breakthrough has occurred, get the time-
        If (Dabs(TmSpan).LE.1.0E-06) TmSpan = (TmStop-Time)/200.0    !spacing needed for the breakthru curve;
        IF (Time-TmLast.GT.TmSpan) Call BrkOut                       !generate a point of the BTC when needed;
      ENDIF
      If (Time.LT.TmStop) THEN                                       !if current time is .ess than the terminate time)
        Call SolMov                                                  !simulate the ion exchange)
        If (.Not.DtAltr.AND.MOD(Irepet,10).EQ.1) Call MbalCk         !periodically output an error report;
      ENDIF
      FlowOu(0) = FlowOu(0) + Dt*ConSat                              !track water movement out of the column)
C-------UNTIL---------------------------------------
      IF (Time.LT.TmStop) GOTO 30                                    !until the terminate pore volume is attained;
      Call BrkRep                                                    !produce the break-through report;
      Call RepItr                                                    !report the iterative trace percentages)
      Call RepExt                                                    !report the min/max values)
      If (NdtSum.GT.0) Write(12,1040) NdtSum,DtSum/NdtSum            !report the average Dt value;
 1040   Format(/,' Dt values used: ',I5,' Average Dt: ',G12.5)
      CLOSE(12)                                                      !close the report unit & the profile unit;
      Close(14)
      END
      SUBROUTINE SolMov
C-----------------------------------------------------------------------------------------------------------------
C This routine calls the routines that perform the simulation and coordinates the sequence of calculations.
C-----------------------------------------------------------------------------------------------------------------
      INCLUDE 'Hydrazin.CMB'
      Logical RvDone
      Dimension SolInp(MaxIon)                                      !this vector stores the current input concentrations
      Do 10 Ion = 1,Nions                                            !for all ions, transfer the input concentrations
        SolInp(Ion) = FluxAm(IFlxPt,Ion)                             !into SOLINP.  If going into a flux change,
        If (LckDwn.LE.10) SolInp(Ion) = FluxAm(IFlxPt-1,Ion) +       !ease into flux change by 10% steps;
     &       (FluxAm(IFlxPt,Ion)-FluxAm(IFlxPt-1,Ion))*LckDwn/10.0
        If (Irepet.GE.20) Call MinMax(Errors(Ion,1),Ion,1)           !monitor min/max errors after 20 iterations;
        DO 10 IDEP = 1,Nodes                                         !for all nodes in the column ..)
          If (Ion.GE.2.AND.Ion.LE.3) THEN                            !and .or ions in the ion exchange (Na+ & Ca++);
            OldSor(Ion,Idep) = Csorb(Ion,Idep)                       !transfer the current values to the OLD arrays &;
            Call MinMax(Csorb(Ion,Idep),Ion,3)                       !update min/max for the sorbed compartment;
          ENDIF
          If (Ion.LE.2) THEN                                         !transfer hydrazin(e)(ium) sorbed materials
            OldAds(Ion,Idep) = Adsorb(Ion,Idep)
            OldFlx(Ion,Idep) = FlxSrb(Ion,Idep)
            OldBnd(Ion,Idep) = Bound(Ion,Idep)
          ENDIF
          Call MinMax(CDissv(Ion,Idep),Ion,2)                        !do the same for min/max values for all ions &;
   10     OldDis(Ion,IDEP) = CDissv(Ion,IDEP)                        !move the current values set into the 'OLD' arrays;
      DtSum = DtSUm + Dt                                             !track the DT value)
      NDtSum = NDtSum + 1
```

```
      Ipass = 0
      If (Hydrol) Call ConInp(SolInp(1),SolInp(2))                    {if using hydrolysis, do it on the input values}
      RvDone = .false.                                                {= reversible sorption not yet calculated}
   15 ConVrg = 0.0                                                    {holds the maximum error between sucessive trials}
      Do 20 Ion = 1,Nions
   20   If (UseIon(Ion)) Call SolIon(Ion,SolInp(Ion),Convrg,Ipass)    {solve the transport equations for all ions}
      If (GoIonX) Call Exchang                                        {perform ion-exchange and for the N2H4 & N2H5+...}
      Ipass = Ipass +1
      If (Revers.AND.(Kinetc(1).OR.Kinetc(2))) THEN                  {if doing reversible sorption & at least 1 hydrazine}
        Call RevAds                                                   {is kinetic, calculate the reversible effect & set}
        RvDone = .true.                                               {the flag to true}
      ENDIF
      If (Ipass.EQ.1.OR.                                             {it's the 1st pass or the passes have not exceeded the}
     &      (Ipass.LT.ItrLim.AND.Convrg.GT.Tlevel)) Goto 15           {limit but no convergence yet, so try again...}
      ItrTrk(Ipass) = ItrTrk(Ipass) + 1                               {note the number of iterations needed}
      If (.Not.RvDone.AND.Revers) Call RevAds                         {if needed, do the reversible sorption calculations}
      Do 30 Idep = 2,Nodes-1                                         {for all internal nodes in the column & the hydrazine}
        Do 30 Ion = 1,2
          If (AutoOx) Call OxyEat(Cdissv(Ion,Idep),OxNitr(Idep),     {species, allow autoxidation & hydrolysis if being used}
     &                Cdissv(4,Idep),Ion,Idep)
   30     If (Hydrol)Call Conver(Cdissv(1,Idep),Cdissv(2,Idep))
      Do 45 Idep = 2,Nodes-1                                          {for mass balance, calculate the amount sorbed }
        DO 45 Ion = 1,2                                               {irreversibly}
   45     SrbOut(Ion) = SrbOut(Ion) + (RatIrv(Ion) + Rmicrb(Ion,1)) *
     &     Dt * (Cdissv(Ion,Idep)+OldDis(Ion,Idep))*0.5 * Dx * ThetaS
      DO 50 Ion = 1,Nions                                             {MASS BALANCE CALCULATIONS...}
        FLOWIN(Ion) = FLOWIN(Ion) + FluxAm(Iflxpt,3)* SolInp(Ion) * DT  {the amount that has entered is the flux * DT}
        AvConc = (Cdissv(Ion,Nodes-1) + Cdissv(Ion,Nodes) +           {the average concentration across the bottom}
     &            OldDis(Ion,Nodes-1) + OldDis(Ion,Nodes))/4.0
   50   FLOWOU(Ion) = FLOWOU(Ion) + ConSat * DT * AvConc             {amount that left = flux * dt * average}
      RETURN
      END
      Double Precision Function Rkintc(Idep,Ion)
C-----------------------------------------------------------------------------------------------------------------------
C This routine handles reversible sorption for the kinetic version
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE 'Hydrazin.CMB'
      Rkintc = 0.0
      If (Revers.AND.Ion.LE.2.AND.                                    {if doing reversible sorption, for a hydrazine spp.}
     &      RatRev(Ion,1)*RatRev(Ion,2)*RatRev(Ion,3).GT.Apprx0) THEN  {with positive rate coefficents & kinetically....}
        If (Kinetc(Ion)) Rkintc =
     &       Dt * (-RatRev(Ion,1) * CDMean(Ion,Idep)**RatRev(Ion,3) +   {set up the appropriate factor}
     &              BulkDn * RatRev(Ion,2) * AdMean(Ion,Idep) / ThetaS)
      ENDIF
      Return
      END
      Double Precision Function RequlK(Idep,Ion)
C-----------------------------------------------------------------------------------------------------------------------
C This routine handles reversible sorption - Equilibrium Version
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE 'Hydrazin.CMB'
      RequlK = 0.0
      If (Revers.AND.Ion.LE.2.AND.                                    {if doing reversible sorption, for a hydrazine spp}
     &      Requil(Ion)*RatRev(Ion,3).GT.Apprx0) THEN                 {with positive equilibrium rate coefficients...}
        If (.not.Kinetc(Ion)) RequlK =
     &       BulkDn * RatRev(Ion,3) * Requil(Ion) *                   {set up the appropriate factor}
     &       CDmean(Ion,Idep)**(RatRev(Ion,3)-1.0) / ThetaS
      ENDIF
      Return
      END
      Double Precision Function DecayI(Ion)
C-----------------------------------------------------------------------------------------------------------------------
C This routine handles Irreversible Decay
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE 'Hydrazin.CMB'
      If (Ion.LE.2.AND.Irrevs) THEN                                   {if doing irreversible sorption for a hydrazine}
        DecayI = Dt * RatIrv(Ion)                                     {spp, set up the factor}
      ELSE
        DecayI = 0.0
      ENDIF
      Return
      END
      Double Precision Function DecayM(Ion)
C-----------------------------------------------------------------------------------------------------------------------
C This routine handles Microbial decay
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE 'Hydrazin.CMB'
      If (Ion.LE.2.AND.Irrevs) THEN                                   {see comments for DECAYI}
        DecayM = Dt * Rmicrb(Ion,1)
      ELSE
        DecayM = 0.0
      ENDIF
      Return
      END
      SUBROUTINE SolIon(Ion,ConcIn,Convrg,Ipass)
C-----------------------------------------------------------------------------------------------------------------------
C This routine calculates the ion transport equation using a finite-difference, tri-diagonal approach.
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE 'Hydrazin.CMB'
      Dfactr =  Disper*Dt/(2.0*Dx*Dx)                                 {set up these factors outside of the loop}
      Vfactr =  Veloc*Dt/(4.0*Dx)
      DO 10 IDEP = 2,Nodes - 1                                        {for the internal nodes within the column}
        CALL GHfixd(IDEP,Ion,Eta,DIFF)                                {get the ion exchange coefficients}
        RSorbE = RequlK(Idep,Ion)                                     {the reversible equilibrium or kinetic values}
        RsorbK = Rkintc(Idep,Ion)
        Decay =  DecayI(Ion)/2.0 + DecayM(Ion)/2.0                    {the 1st order degradation coefficients}
        BndFir = 0.0                                                  {generally the finite sink factor is 0 but if the}
        If (Ion.LE.2) THEN                                            {ion is a hydrazine spp and there is unused capacity}
          If (BdOpen(Ion,Idep)) BndFir =                              {at this node, calculate the transfer factor}
     &                          Dt*BndFac(Ion,Idep)*RatRev(Ion,7)/2.0
        ENDIF
```

```
      SbDiag(IDEP) = -Dfactr - Vfactr                                    {set up the standard tri-diagonal scheme}
      Diagon(IDEP) = 1.0 + Eta + RsorbE + Decay + 2.0*Dfactr + BndFir
      SpDiag(IDEP) = -Dfactr + Vfactr
      Result(IDEP) =
     &   OldDis(Ion,IDEP-1)*(Dfactr + Vfactr) +
     &   OldDis(Ion,IDEP  )*(1.0+Eta+RsorbE-Decay-2.0*Dfactr-BndFir) +
     &   OldDis(Ion,IDEP+1)*(Dfactr - Vfactr) + RsorbK + Diff
   10 If (Result(Idep).LT.0.0) Result(Idep) = 0.0                        {insure no negative concentrations will occur}
      SbDiag(1) = 0.0D+00                                                {FLUX BOUNDARY at the top}
      Diagon(1) =  Disper/Dx + Veloc/2.0
      SpDiag(1) = -Disper/Dx + Veloc/2.0
      Result(1) =  Veloc * ConcIn
      SbDiag(Nodes) = -1.0                                               {REFLECTION BOUNDARY at Bottom}
      Diagon(Nodes) = 1.0
      SpDiag(Nodes) = 0.0D+00
      Result(Nodes) = 0.0D+00
      CALL TriDim                                                        {solve system of equations}
      DO 20 IDEP = 1,Nodes                                               {filter the results}
        IF (Result(IDEP).GT.Apprx0) THEN                                 {if the results are greater than 0...}
          If (Ipass.GT.0.AND.Idep.GE.2.AND.Idep.LE.Nodes-1) THEN         {and this is not the first iteration...}
            If (Cdissv(Ion,Idep).GT.1.0E-20) THEN                        {for values large enough to be worried about...}
              Error = 100.0*DABS((Result(Idep)- Cdissv(Ion,Idep))/       {calculate the error between estimations}
     &                           Cdissv(Ion,Idep))
            ELSE
              Error = 0.0
            ENDIF
            If (Error.GE.Convrg) THEN                                    {if it is larger than the max, change the max}
              Convrg = Error
              MxIon = Ion
              MxDep = Idep
            ENDIF
          ENDIF
          Cdissv(Ion,IDEP) = RESULT(IDEP)                                {and assign the new value into the array}
        ELSE
          Cdissv(Ion,IDEP) = Apprx0                                      {otherwise..set to effective zero}
        ENDIF
   20 Continue
      RETURN
      END
      Subroutine RevAds
C-----------------------------------------------------------------------------------------------------------------------------
C This routine calculates the reversible sorption arrays
C-----------------------------------------------------------------------------------------------------------------------------
      INCLUDE 'Hydrazin.CMB'
      Do 30 Ion = 1,2                                                    {for the hydrazine species...}
        If (UseIon(Ion)) THEN                                            {which are in use...}
          If (.Not.Kinetc(Ion)) THEN                                     {and are in an equilibrium model...}
            Do 10 Idep = 1,Nodes
              Adsorb(Ion,Idep)=Requil(Ion)*                              {calculate the amount adsorbed...}
     &                Cdissv(Ion,Idep)**RatRev(Ion,3)
              BnFact = BndFac(Ion,Idep)
              FixSrb(Ion,Idep) = RatRev(Ion,4)*Cdissv(Ion,Idep)/
     &           RatRev(Ion,5) - RatRev(Ion,6)*BnFacT*OldFix(Ion,Idep)
   10         Bound(Ion,Idep) = RatRev(Ion,6)*BnFacT*OldFix(Ion,Idep)+
     &                           OldBnd(Ion,Idep)
          ELSE                                                           {OTHERWISE---KINETIC Model---}
            Do 20 Idep = 2,Nodes-1
              If (RatRev(Ion,1)*RatRev(Ion,2).GT.Appx0) THEN             {if the rates to loosely sorbed (LS) are >0}
                BnFact = BndFac(Ion,Idep)
                DisFac = Dt*ThetaS*RatRev(Ion,1)*                        {=amount moving into LS from dissolved phase}
     &                     Cdmean(Ion,Idep)**RatRev(Ion,3)/BulkDn
                ComRat = Dt*(RatRev(Ion,2)+RatRev(Ion,4)+                {=amount leaving LS into tightly sorbed (TS)}
     &                     RatRev(Ion,6)*BnFact        )/2.0             { permanently bound (PB) and dissolved phase}
                FixFac = Dt*RatRev(Ion,5)*FxMean(Ion,Idep)               {=amount entering LS from TS}
                AdsFac = (1.0-ComRat)*OldAds(Ion,Idep)
                Adsorb(Ion,Idep) = (DisFac+AdsFac+FixFac)/(1.0+ComRat)   {combine into new LS (adsorbed) concentration}
              ENDIF
              If (RatRev(Ion,4)*RatRev(Ion,5).GT.Appx0) THEN             {if the rates to/from TS are > 0,...}
                ComRat = 0.5*Dt*RatRev(Ion,5)
                Fxfac1 = Dt*RatRev(Ion,4)*AdMean(Ion,Idep)               {=amount entering & amount leaving}
                FxFac2 = (1.0-ComRat)*OldFix(Ion,Idep)
                FixSrb(Ion,Idep) = (FxFac1 + FxFac2)/(1.0 + ComRat)      {determine the new conc. in TS compartment}
                If (FixSrb(Ion,Idep).LE.Apprx0) THEN
                  Write(12,2000) Ion,Idep,FixSrb(Ion,Idep)
 2000               Format('ERROR: Chemi-Sorbed over-reduced:Fx(',2I3,
     &                  ')=',G12.5,'. Pgm ends.')
                  Call EndPgm('Chemi-Sorbed < Apprx0!')
                ENDIF
              ENDIF
              If (BdOpen(Ion,Idep).AND.Sponge.GT.0.0) THEN               {if there is open capacity & permanent binding}
                FromCp = Dt*RatRev(Ion,6)*AdMean(Ion,Idep)               {can occur, get the amount from either TS or }
                FromCd = Dt*RatRev(Ion,7)*ThetaS*CdMean(Ion,Idep)        {LS - NOTE either RATREV(ION,6) or }
     &                           /BulkDn                                 {RATREV(ION,7) must be zero}
                ComRat = (FromCp + FromCd)/(2.0*Sponge)
                Bound(Ion,Idep) = (FromCp +FromCd +                      {calculate the new bound quantity and set the}
     &             OldBnd(Ion,Idep)*(1.0-ComRat))/(1.+ComRat)            {boolean of available capacity appropriately}
                If (Bound(Ion,Idep).GE.Sponge) THEN
                  Bound(Ion,Idep) = Sponge
                  BdOpen(Ion,Idep) = .false.
                ENDIF
              ENDIF
   20       Continue
          ENDIF
        ENDIF
   30 Continue
      Return
      END
      Double Precision Function BndFac(Ion,Idep)
C-----------------------------------------------------------------------------------------------------------------------------
C This routine calculates the scale factor for the sink = (Maximum possible - Current_Value)/Maximum possible
C-----------------------------------------------------------------------------------------------------------------------------
```

```
      INCLUDE 'Hydrazin.CMB'
      If (Sponge.GT.0.0) THEN                                   {if there is a set max sink...}
        BndFac = (Sponge-BnMean(Ion,Idep))/Sponge               {the capacity left is (max-current)/max;
        If (BndFac.GT.1.0) BndFac = 1.0                         {which must be between 0 and 1}
        If (BndFac.LT.0.0) BndFac = 0.0
      ELSE
        BndFac = 1.0                                            {otherwise return 1}
      ENDIF
      Return
      END
      Subroutine Conver(Hydine,Hydium)
C-----------------------------------------------------------------------------------------------------------------------
C This routine converts hydrazine to hydrazium and back again.  If Ambient pH is ever not a column constant, then the
C nodal argument must be added to this routine's argument list
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE 'Hydrazin.CMB'
      Ratio = 10.0**(BasepH-AmbnpH(1))                          {determine the conversion ratio & the total}
      Total = Hydine + Hydium
      Call AdjstH(Hydine,Total/(Ratio+1.0),1)                   {adjust each as a function of the ratio & total}
      Call AdjstH(Hydium,Total-Hydine,2)
      Return
      END
      Subroutine AdjstH(HOld,Hnew,Ion)
C-----------------------------------------------------------------------------------------------------------------------
C This routine adjusts a hydrazin(e)(ium) value & flowin & flowou for mass balance control
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE 'Hydrazin.CMB'
      Hdiff = HOld - HNew
      If (Hdiff.GT.Apprx0) THEN
        FlowOu(Ion) = FlowOu(Ion) + Hdiff*Dx
        HydrlA(Ion) = HydrlA(Ion) + Hdiff*Dx
      ELSE
        FlowIn(Ion) = FLowIn(Ion) - Hdiff*Dx
        HydrlA(Ion) = HydrlA(Ion) - Hdiff*Dx
      ENDIF
      HOld = HNew
      Return
      END
      Subroutine ConInp(Hydine,Hydium)
C-----------------------------------------------------------------------------------------------------------------------
C This routine converts hydrazine to hydrazium and back again. If Ambient pH is ever not a column constant, then the
C nodal argument must be added to this routine's argument list. Used for column Input ONLY
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE 'Hydrazin.CMB'
      Ratio = 10.0**(BasepH-AmbnpH(1))
      Total = Hydine + Hydium
      Call AdjInH(Hydine,Total/(Ratio+1.0),1)
      Call AdjInH(Hydium,Total-Hydine,2)
      Return
      END
      Subroutine AdjInH(HOld,Hnew,Ion)
C-----------------------------------------------------------------------------------------------------------------------
C This routine adjusts a hydrazin(e)(ium) value
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE 'Hydrazin.CMB'
      Hdiff = HOld - HNew
      HydrIn(Ion) = HydrIn(Ion) + Hdiff*Dt*FluxAm(IflxPt,0)
      HOld = HNew
      Return
      END
      SUBROUTINE GHFixd(IDEP,Ion,Eta,DIFF)
C-----------------------------------------------------------------------------------------------------------------------
C This routine generates the coefficients for relating the sorbed phase to the dissolved phase for fixed K's
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE 'Hydrazin.CMB'
      Dimension SorVal(MaxIon),DisVal(MaxIon)
      DIFF  = 0.0                                               {initialize accumulators}
      Eta = 0.0
      If (GoIonX.AND.Ion.GE.2.AND.Ion.LE.3.AND.CECfix.GT.0.0) THEN
          HFactr = 0.0                                          {otherwise apply the Valocchi technique}
          GFactr = 0.0
          SMCSGB = 0.0D+00
          Do 10 I = 2,3                                         {define dissolved & sorbed values at 'N+1/2'}
            SorVal(I)=0.5*( Csorb(I,Idep) + OldSor(I,IDep))
   10       DisVal(I)=0.5*(Cdissv(I,Idep) + OldDis(I,Idep))
          DO 20 J = 2,3
            IF (J.NE.Ion) THEN                                  { hij*(CSj[N+1]-CSj[N])}
              DIFF = DIFF + (Cdissv(J,IDEP)-OldDis(J,IDEP)) *   { & sum  Vj*CS(jk) }
     &                      (SorVal(J)/DisVal(J))
              SMCSGB = SMCSGB + VALENC(J)*SorVal(J)
            ENDIF
   20     CONTINUE
          GFactr = 1.+ SMCSGB/(VALENC(Ion) * SorVal(Ion))       {calculate Gii }
          HFactr = SMCSGB / (VALENC(Ion)*DisVal(Ion))           {calculate Hii }
          Tfactr = ThetaS*Gfactr
          Diff   = -1.0*BulkDn*Diff/Tfactr
          Eta    =  BulkDn*HFactr/TFactr
        ENDIF
      Return
      END
      SUBROUTINE Exchang
C-----------------------------------------------------------------------------------------------------------------------
C This routine calculates the sorbed concentrations for all ions at all nodes
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE 'Hydrazin.CMB'
      If (CECFix.GT.0.0) THEN
        Do 10 Idep = 2,Nodes-1
          Omega = SCa2Hy/(CECfix*Cdissv(2,Idep)**2)             {use the quadratic equation to solve the +1 valence
          Csorb(2,Idep) = (-1.0 + DSQRT(1.0 + 4.0*CECfix*Omega))/
     &                                (2.0*Omega)               {and filter for minimum value}
          If (Csorb(2,Idep).LT.Apprx0) Csorb(2,Idep)=Apprx0
```

```
        Csorb(3,Idep) = CECfix - Csorb(2,Idep)                    {determine the 2nd by difference and filter the value;
   10   If (Csorb(3,Idep).LT.Apprx0) Csorb(3,Idep)=Apprx0
      ENDIF
      Return
      END
      Subroutine OxyEat(Hconc,Complx,OxConc,Ion,Idep)
C-----------------------------------------------------------------------------------------------------------------------
C This routine implements the auto-oxidation aspects. RautoX(1,)=Hydrazine;  2,= hydrazinium; RautoX(,1)=to complex; ,
C 2= from complex; ,3=to water - HAS NOT BEEN TESTED!!!
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE 'Hydrazin.CMB'
      If (Hconc.GT.Apprx0.AND.OxConc.GT.Apprx0) THEN
        RequlX = RautoX(Ion,1)/(RautoX(Ion,1) + RautoX(Ion,2))
        If (Hconc.GT.OxConc) THEN
          Cleft = Oxconc
        ELSE
          Cleft = Hconc
        ENDIF
        ToComx = RautoX(Ion,1)*Dt*Cleft
        ToLeft = RautoX(Ion,2)*Dt*Complx
        Complx = Complx - ToLeft + ToComx
        ToWatr = RautoX(Ion,3)*Dt*Complx
        Complx = Filter(Complx - ToWatr,Apprx0,1.0E+30)
        ConcNw = Filter(Hconc - ToComx + ToLeft,Apprx0,1.0E+30)
        OxConc = Filter(OxConc - ToComx + ToLeft,Apprx0,1.0E+30)
        Diff =  (Hconc-ConcNw)*Dx*ThetaS
        If (Diff.GT.Apprx0) SrbOut(Ion) = SrbOut(Ion) + Diff
        Hconc = ConcNw
      ENDIF
      Return
      END
      SUbroutine MinMax(Val,Ion,Index)
C-----------------------------------------------------------------------------------------------------------------------
C This routine monitors the minimum and maximum values for all ions for three parameters
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE 'Hydrazin.CMB'
      Logical FirPas(MaxIon,0:4)
      Data FirPas/20*.true./
      If (FirPas(Ion,Index)) THEN
         Extrem(1,Ion,Index) = Val
         Extrem(2,Ion,Index) = Val
         FirPas(Ion,Index) = .false.
      ELSE
        If (Val.LT.Extrem(1,Ion,Index)) Extrem(1,Ion,Index) = Val
        If (Val.GT.Extrem(2,Ion,Index)) Extrem(2,Ion,Index) = Val
      ENDIF
      Return
      END
      Subroutine RepExt
C-----------------------------------------------------------------------------------------------------------------------
C This routine reports the min/max data fro all ions for 3 parameters
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE 'Hydrazin.CMB'
      Character Types(0:3)*20
      Data Types/' ','Mass Balance','Dissolved','Sorbed'/
      If (CECfix.Gt.1.0E-06) THEN
        Ntype = 3
      ELSE
        Ntype = 2
      ENDIF
      Do 10 I = 1,Ntype
        Write(12,1000) Types(I)
 1000    Format(//,' Extremes of ',A)
        Do 10 Ion = 1,Nions
   10     If (UseIon(Ion)) Write(12,1010) Ion,Solute(Ion),
     &           (Extrem(J,Ion,I),J=1,2)
 1010      Format(10X,I2,1X,A,' Minimum=',G12.5,' Maximum=',G12.5)
      Return
      END
      Double Precision Function Filter(Value,ValBot,ValTop)
C-----------------------------------------------------------------------------------------------------------------------
C This function insures than any value will fall between 0 and 1
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE 'Hydrazin.CMB'
      If (Value.LT.ValBot) THEN
        Filter = ValBot
      ELSE If (Value.GT.ValTop) THEN
        Filter = ValTop
      ELSE
        Filter = Value
      ENDIF
      Return
      END
      Subroutine StartR
C-----------------------------------------------------------------------------------------------------------------------
C This routine starts the simulation run by initization & reading the parameter file
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE 'Hydrazin.CMB'
      Logical NoFixd
      Character Aline*80
      Write(*,1000)
 1000   Format('SPlease enter Parameter filename: ')              {get the info from the user}
      Read(*,1010) Fprint
 1010   Format(A)
      Irepet = 0                                                {set variables to 0}
      TmLast = 0.0
      TmSpan = 0.0
      Valenc(2) = 1.0                                           {insure that the valences are set}
      Valenc(3) = 2.0
      Write(*,*) 'Parameter file will be ',Fprint
    5 OPEN(UNIT=10,FILE=Fprint,STATUS='OLD',ERR=100)
```

```
      Call GetPar                                                    (read the run parameters)
      If (.Not.UseIon(4)) THEN
        NinUse = 3
      ELSE
         NinUse = 4
      ENDIF
      Ion1st = 0
      Ion1st = 1
      If (.not.UseIon(1).AND.Ion1st.EQ.1) Ion1st = 2
      Do 10 I = 1,30
   10   ItrTrk(I) = 0
      Itmptr = 0
      DtStar = .true.
      NDtSum = 0
      DtSum = 0.0
      DtMin = 0.5*Dx*Dx*CapMin/ConSat                                (stability criterion establishes floor)
      IF (DtMin.LT.0.1) DtMin = 0.1
      Write(*,1020) DtMin
 1020   Format(' Dtmin is set to ',G12.5)
      IflxPt = 0
      Do 15 I = 0,Nions
   15   FluxAm(0,I) = FluxAm(1,I)
      PoreVl = 0.0
      If (GoIonX) THEN
        Write(*,*) '---Generating initial ion-exchange equilibrium---'
        Call Exchang
        Do 20 Idep = 1,Nodes
          Do 20 Ion = 2,3
   20       OldSor(Ion,IDep) = Csorb(Ion,Idep)
      ENDIF
      If (Revers) THEN
        Write(*,*) '---Generating initial reversible equilibrium---'
        Do 25 Ion = 1,2
          If (RatRev(Ion,1)*RatRev(Ion,2).LE.Apprx0) THEN
            Requil(Ion) = 0.0
          ELSE
            Requil(Ion) = ThetaS*RatRev(Ion,1)/(RatRev(Ion,2)*BulkDn)
          ENDIF
          If (RatRev(Ion,4)*Ratrev(Ion,5).LE.Apprx0) THEN
            NoFixd = .true.
          ELSE
            NoFixd = .false.
          ENDIF
          Do 25 Idep = 1,Nodes
            If (Requil(Ion).GT.Apprx0) THEN
              Adsorb(Ion,Idep) = Requil(Ion)*Cdissv(Ion,Idep)**
     &                           RatRev(Ion,3)
              If (Adsorb(Ion,Idep).LT.Apprx0) Adsorb(Ion,Idep)=Apprx0
            ELSE
              Adsorb(Ion,Idep) = 0.0D+00
            ENDIF
            Bound(Ion,Idep) = 0.0
            OldBnd(Ion,Idep) = 0.0
            BdOpen(Ion,Idep) = .true.
            If (NoFixd) THEN
              FixSrb(Ion,Idep) = 0.0
            ELSE
              FixSrb(Ion,Idep) = RatRev(Ion,4)*Adsorb(Ion,Idep)/
     &                           RatRev(Ion,5)
              If (FixSrb(Ion,Idep).LT.Apprx0) FixSrb(Ion,Idep) = Apprx0
            ENDIF
            OldFix(Ion,Idep) = FixSrb(Ion,Idep)
   25     Continue
      ENDIF
      Do 60 I = 0,Nions                                              (zero the various accumulators)
         FlowIn(I) = 0.0D+00
   60    FLowOu(I) = 0.0D+00
      Do 65 I = 1,Nions
   65   SrbOut(I) = 0.0
      Time =0.0
      DtSpan = (1.0/DtStep + 1)*(DtMax/2.0+DtMin)-DtMin
      Write(*,*) Rlabel
      Do 70 Ion = 1,2
        If (UseIon(Ion)) THEN
          Aline = Solute(Ion)(1:5)//':'//'Diss.'
          If (RatRev(Ion,1)+RatRev(Ion,2).GT.Apprx0)
     &      Aline(12:22)='<-->Phys1-S'
          If (RatRev(Ion,4)+RatRev(Ion,5).GT.Apprx0)
     &      Aline(23:33)='<-->Chemi-S'
          If (RatRev(Ion,6).GT.Apprx0)
     &      Aline(34:42)='>-->Bound'
          If (RatRev(Ion,7).GT.Apprx0)
     &      Aline(38:54)='Bound<-(1)-<Diss.'
          Write(12,*) '     '//Aline
          Write(*,*) Aline
        ENDIF
   70 Continue
      LckDwn = 11
      Goto 110
  100 Call EndPgm('Parameter file not found')
      Return
      END
      SUBROUTINE GetPar
C-------------------------------------------------------------------------------------------------------------------------------
C This routine reads the start parameter file
C-------------------------------------------------------------------------------------------------------------------------------
      INCLUDE 'Hydrazin.CMB'
      Logical AtEOF
      CHARACTER Ftable*15,Fisoth*15,Label*8,Rest*65,Dummy*80
      Read(10,1000) Fprint                                           (read the file name for the output & open that file)
 1000   Format(A)
```

```
      Open(Unit=12,File=Fprint,Status='New')
    FBrkTh = Fprint(1:Iextnt(Fprint))//'BRK'
      Open(Unit=13,File=FBrkth,Status='NEW')                          {open the BTC file}
    Open(Unit=14,File=Fprint(1:Iextnt(Fprint))//'PRF',Status='NEW')   {open the BTC file}
    WRITE(12,1010) Char(12)                                           {write a simulation report heading}
1010  Format(A,31X,'Hydrazin(e)(ium) Transport - VER. 870929 ',
   &  '- DR.S.A.BLOOM',/,1X,'+',128('-'),'+',/,
   &'    This is a finite difference program capable of handl',
   &'ing 4 species in a single soil-layer column. It features:',/,
   &' <1> Dissolved & sorbed phases of Hydrazine, Hydrazinium & Ca',
   &'lcium and  Oxygen concentrations',/,
   &' <2> Hydrolysis; AutoOxidation; Microbial Degradation; Sorpti',
   &'on (Reversible & Reversible); and Ion-Exchange using the ',/,
   &'fixed Valocchi Model',/,
   &' *******---> LAST MAJOR REVISION -- Oct. 12,1987 <---********  ',
   &' DR.S.A.BLOOM FOR DR. R.S.MANSELL - SOIL SCIENCE DEPT,IFAS  ',/,
   &' +',128('-'),'+',/,28X,'SIMULATION PARAMETERS:',/)
    Write(12,1000) Fprint                                             {now read & echo the various parameters}
    Call EchoSt(Rlabel)
    Call EchoRl(Column)
    Call EchoRl(TmStop)
    Call EchoRl(ThetaS)
    Call EchoRl(ConSat)
    Call EchoRl(BulkDn)
    Call EchoRl(Disper)
    Call EchoRl(Sponge)
1030   Format(A,G12.5,A,G12.5)
    Call EchoLo(GoIonX)
    NIons = 4
    Solute(0) = 'Water'
    Solute(1) = 'N2H4'
    Solute(2) = 'N2H5+'
    Solute(3) = 'Ca++'
    Solute(4) = 'Oxygen'
    SCa2Hy = 0.0
    If (GoIonX) THEN
      Call EchoSt(Fisoth)
      Open(Unit=20,File=Fisoth,Status='OLD',Err=300)
      Write(12,1030) ' Selectivity File: '//Fisoth
      Write( *,1030) ' Selectivity File: '//Fisoth
      Read(20,1000) Dummy
      Call StrCap(DUmmy)
      If (Dummy(1:5).EQ.'VALOC') THEN
        Read(20,*,ERR=300) SCa2Hy
        Write(12,1045) 'Selectivity Ca->N2H5+: ',SCa2Hy
        Write(*,1045) 'Selectivity Ca->N2H5+: ',SCa2Hy
1045      Format(10X,A,G12.5)
      ELSE
        Call EndPgm('Valocchi Model must be used')
      ENDIF
      Close(20)
    ENDIF
    Call EchoLo(Revers)
    Do 5 Ion = 1,2
      If (Revers) THEN
        Call EchoLo(Kinetc(Ion))
        If (.Not.Kinetc(Ion)) THEN
          Write(12,*) '  Equilibrium Sorption will be used for ',
   &                            Solute(Ion)
        ELSE
          Write(12,*) '  Kinetic Sorption will be used for ',
   &                            Solute(Ion)
        ENDIF
      ENDIF
      Do 5 Irate = 1,7
        If (Revers) THEN
          Call EchoRl(RatRev(Ion,Irate))
        ELSE
          RatRev(Ion,Irate) = 0.0
        ENDIF
 5  Continue
    Call EchoLo(Irrevs)
      Do 10 Ion = 1,2
        If (Irrevs) THEN
          Call EchoRl(RatIRv(Ion))
        ELSE
          RatIrv(Ion) = 0.0
        ENDIF
10  Continue
    Call EchoLo(Microb)
      Do 15 Ion = 1,2
        Do 15 Irate = 1,4
          If (Microb) THEN
            Call EchoRl(Rmicrb(Ion,Irate))
          ELSE
            Rmicrb(Ion,Irate) = 0.0
          ENDIF
15  Continue
    Call EchoLo(AutoOx)
      Do 20 Ion = 1,2
        Do 20 Irate = 1,3
          If (AutoOx) THEN
            Call EchoRl(RautoX(Ion,IRate))
          ELSE
            RautoX(Ion,IRate) = 0.0
          ENDIF
20  Continue
    Call EchoLo(Hydrol)
    If (Hydrol) THEN
      Call EchoRl(BasepH)
      Call EchoRl(AmbnpH(1))
```

```
        If (AmonpH(1).GT.BasePh) Call EndPgm('pH value > base value')
      ELSE
        BasepH = 0.0
        AmonpH(1) = 0.0
      ENDIF
      Call EchoIn(Nspecs)
      If (Nspecs.GT.Maxset) Call EndPgm('Too many boundary sets')
      Do 50 I = 1,Nspecs
        Call EchoRl(ColDep(I))
        Do 45 J = 1,Nions
          Call EchoRl(ColIon(I,J,1))
          If (ColIon(I,J,1).GT.1.0E-06.AND.J.LE.2.AND.Revers)
     &      Write(*,*) '---WARNING: GT 0 N2H4 or N2H5+ concentration',
     &       '+ reversible sorption should not be used---'
   45     If (J.GE.2.AND.J.LE.3) Call EchoRl(ColIon(I,J,2))
   50 Call EchoRl(CECFix)
      Call EchoRl(GridAm)
      Call EchoRl(Dt)
      DX = GridAM
      NODES = NINT(COLUMN/DX)+2
      If (Nodes.GT.MaxNod) Call EndPgm('Too many nodes')
      DX = COLUMN/(NODES-2)
      DtMin = 0.1
      DtMax = Dt
      Call ColSet
      Call EchoIn(Nflux)
      If (Nflux.GT.Maxset) Call EndPgm('Too many flux sets')
      Do 55 I = 1,Nflux
        Call EchoRl(FluxTm(I))
        Call EchoRl(FluxAm(I,0))
        FluxAm(I,0) = FLuxAm(I,0)/3600.0
        Write(12,1045)'Imposed flux --> cm/sec=',FluxAm(I,0)
        If (I.GT.1.AND.FluxAm(I,0).NE.FluxAm(1,0)) Call EndPgm(
     &   'Changing fluxes are not allowed in a steady flow system')
        Do 55 Ion = 1,Nions
   55     Call EchoRl(FluxAm(I,Ion))
      If (ConSat.NE.FluxAm(1,0)) THEN
        ConSat = FluxAm(1,0)
        Write(12,1045)'Saturated Conductivity reset to ',Consat
      ENDIF
      Veloc = ConSat/ThetaS
      Call EchoIn(Nrepr)
      If (Nrepr.GT.Maxrep) Call EndPgm('Too many reports requested')
      Do 60 I = 1,Nrepr
   60   Call EchoRl(TmRepr(I))
      IF (TMREPR(NREPR).LT.TmStop) THEN
        NREPR = NREPR + 1
        TMREPR(NREPR) = Tmstop
      ENDIF
      TmRepr(0) = 0.0
      Do 61 I = 1,Nions
   61   Call EchoRl(Scale(I))
      Call EchoLo(FeedBk)
      Call EchoIn(Icheck)
      Call EchoRl(Tlevel)
      Call EchoIn(ItrLim)
      If (ItrLim.GT.30) Call EndPgm('Iterative Limit must <= 31')
      Call EchoRl(DtStep)
      Call EchoRl(FloorP)
   65 Close(10)
      Write(12,1050) Icheck,TLevel,ItrLim,DtStep,FloorP
 1050  Format(' External abort check      = ',I12,/,
     &       ' Iterative Tolerance Limit = ',G12.5,' ( %)',/,
     &       ' Iterative Pass Maximum    = ',I12,/,
     &       ' Step Amount for Dt change = ',G12.5,' (frac)',/,
     &       ' Effective Zero Value power= ',G12.5)
      ApprxO = 1.0*10**FloorP
      RETURN
  300 Write(*,*) 'Isotherm file: '//Fisoth//' not found!. Pgm Aborts.'
      Stop
      END
      SUBROUTINE DepBrk (Depth,NsetAb,Matchs)
C-------------------------------------------------------------------------------------------------------------------------------
C This routine determines the node for any given depth in the column
C-------------------------------------------------------------------------------------------------------------------------------
      INCLUDE 'Hydrazin.CMB'
      Logical Matchs
      Matchs = .true.
      IF (Depth.LT.ColDep(1) ) THEN
         NsetAB = 1
      ELSE IF (Depth.GT.ColDep(Nspecs)) THEN
         NsetAb = Nspecs
      ELSE IF (Depth.GE.ColDep(1).AND.Depth.LE.ColDep(Nspecs)) THEN
         Matchs = .false.
         Do 10 J = 1,Nspecs                                        (search all specified levels for bracketing
            IF (Depth.GT.ColDep(J).AND.Depth.LT.ColDep(J+1)) Goto 30   ( depths - here bracketed
            IF (Depth.EQ.ColDep(J)) Goto 20                           ( unless an exact match is possible
   10    Continue
   20    Matchs = .true.
   30    NsetAb = J
      ENDIF
      Return
      END
      SUBROUTINE ColSet
C-------------------------------------------------------------------------------------------------------------------------------
C This routine sets the column according to the input controls
C-------------------------------------------------------------------------------------------------------------------------------
      INCLUDE 'Hydrazin.CMB'
      Dimension ConcIn(MaxNod)
      Character Datype(2)*4
      Logical Exact
```

```
      Data DaType/'Diss','Sorp'/
      Do 40 Ion = 1,Nions
        UseIon(Ion) = .false.
        Do 40 Ispec = 1,2
          DO 20 I = 1,Nodes
            Depth = (I-1.5) * Dx                                    {determine depth from surface
             Call DepBrk (Depth, J, Exact)
             IF (Exact) THEN
               ConcIn(I) = ColIon(J,Ion,Ispec)
             ELSE
               Xdiff = ColDep(J+1) - ColDep(J)                      {following is for bracketing interpolation
               PSlope = (ColIon(J+1,Ion,Ispec)-ColIon(J,Ion,Ispec))
     &                                                 /Xdiff       {slope of conc between specif.levels
               PInter = ColIon(J+1,Ion,Ispec) - Pslope * ColDep(J+1)  {intercept of conc line
               ConcIn(I) = PInter + PSlope * Depth                  {linear interpolation of conc values
            ENDIF
            If (DABS(ConcIn(I)).GT.1.0E-30) UseIon(Ion) = .true.
   20     Continue
          IF (Ispec.EQ.1) THEN
            DO 30 I = 1,Nodes
              OldDis(Ion,I) = ConcIn(I)
   30         CDissv(Ion,I) = ConcIn(I)
          ELSE IF (Ispec.EQ.2) THEN
            If (Ion.GE.2.AND.Ion.LE.3) THEN
              DO 35 I = 1,Nodes
                Csorb(Ion,I) = ConcIn(I)
   35           OldSor(Ion,I) = ConcIn(I)
            ELSE
              Do 36 I = 1,Nodes
                Csorb(Ion,I) = 0.0
   36           OldSor(Ion,I) = 0.0
            ENDIF
          ENDIF
   40 Continue
      Do 50 Ion = 1,Nions
        Csorb(Ion,1)      = 0.0
        Csorb(Ion,Nodes)  = 0.0
        OldSor(Ion,1)     = 0.0
   50   OldSor(Ion,Nodes) = 0.0
      Do 60 I = 1,Nodes
        AmbnpH(I) = AmbnpH(1)
        Do 60 Ion = 1,2
          Adsorb(Ion,I) = 0.0
   60     OldAds(Ion,I) = 0.0
      Return
      END
      SUBROUTINE TRIDIM
C-----------------------------------------------------------------------------------------------------------------------
C This routine solves a tri-diagonal matrix
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE 'Hydrazin.CMB'
      DIMENSIon A(MaxNod),BETA(MaxNod),Y(MaxNod)
      A(1) = DIAGON(1)
      BETA(1)  = SPDIAG(1)/A(1)
      Y(1)     = RESULT(1)/A(1)
      DO 201 I = 2,Nodes
        A(I)  = DIAGON(I) - SBDIAG(I) * BETA(I-1)
        BETA(I)   = SPDIAG(I)/A(I)
  201   Y(I)      =(RESULT(I)-SBDIAG(I)*Y(I-1))/A(I)
      Result(Nodes) = Y(Nodes)
      DO 203 I = 1,Nodes - 1
        J = Nodes -I
  203   Result(J) = Y(J) - BETA(J) * Result( J+1)
      RETURN
      END
      SUBROUTINE BRkOut
C-----------------------------------------------------------------------------------------------------------------------
C This routine appends data to the breakthrough file.
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE 'Hydrazin.CMB'
      Dimension Divsor(MaxIon)
      Cnorml = Ctotal(Nodes-1)                                      {determine current normality at the bottom}
      Do 10 Ion = 1,Nions
        If (UseIon(Ion).AND.DABS(Scale(Ion)).GT.1.0E-06) THEN       {if using this ion & there is a scale value}
          Divsor(Ion) = Scale(Ion)                                  {then use that value, otherwise use normality}
        ELSE
          Divsor(Ion) = Cnorml
        ENDIF
   10 Continue
      Write(13,1400) PoreVl,(Cdissv(J,Nodes-1)/Divsor(J)
     &               ,J=1,Nions)
 1400   Format(7G12.5)
      TmLast = Time                                                 {note the time of the last BTC point generation}
      Return
      END
      SUBROUTINE BrkRep
C-----------------------------------------------------------------------------------------------------------------------
C This routine reads the backup Breakthrough file and converts the data into a plot data file
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE 'Hydrazin.CMB'
      DIMENSIon TMOUT(200),BRKTHR(MaxIon,200),Xout(201),Yout(201)
      Character Label*8,Rest*65
      Logical NotEqu
      Close(13)
      OPEN(UNIT=13,FILE=FBrkth,STATUS='OLD')                        {the file will be opened and the data-set read}
      IBRK = 0
   10 Ibrk = Ibrk + 1
        Read(13,1000,END=20) TmOut(Ibrk),(BrkThr(I,Ibrk),I=1,Nions)  {read in the time & the concentrations}
 1000     Format(7G12.5)
        IF (IBRK.LT.200) Goto 10                                    {up to 201 observations}
        Ibrk = Ibrk + 1
```

```
   20 Ibrk = IBrk - 1
      If (Ibrk.GT.0) THEN
        Close(13)                                        (the file will then be discarded and the data recorded)
        OPEN(UNIT=13,FILE=FBrkth,STATUS='NEW')             (-> new file of same name but different format)
        Rest = Rlabel(1:65)
        DO 50 Ion = 1,Nions
          If (UseIon(Ion)) THEN
            Nobs = 1
            Xout(1) = TmOut(1)
            Yout(1) = BrkThr(Ion,1)
            Do 30 I = 2,Ibrk-1
             If(NotEqu(BrkThr(Ion,I-1),BrkThr(Ion,I),BrkThr(Ion,I+1)))
     &            THEN
               Nobs = Nobs + 1
               Xout(Nobs) = Tmout(I)
               Yout(Nobs) = BrkThr(Ion,I)
             ENDIF
   30       Continue
            Nobs = Nobs + 1
            Xout(Nobs) = Tmout(Ibrk)
            Yout(Nobs) = BrkThr(Ion,Ibrk)
            Label = Solute(Ion)(1:8)
            Call WritDP(Label,Rest,Nobs,Xout,Yout,13)
          ENDIF
   50   Continue
      ELSE
        Rewind(13)
        Write(13,*) 'No Data Available'
      ENDIF
        Close(13)
      Return
      END
      Double Precision Function Ctotal(Idep)
C-----------------------------------------------------------------------------------------------------------------------
C this routine calculates normality at a given node
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE 'Hydrazin.CMB'
      Ct = 0.0
      DO 10 Ion = 1,3
   10   Ct = Ct + Cdissv(Ion,Idep)
      Ctotal = Ct
      Return
      END
      SUBROUTINE FilVal(Avalue,ValNew,ValOld)
C-----------------------------------------------------------------------------------------------------------------------
C This routine filters value for divisions by 0 during error calculations
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE 'Hydrazin.CMB'
      Character Avalue*12
      IF (ValOld.GT.Apprx0) THEN
        Call SetChr(Avalue, 100.0*(ValNew-ValOld)/ValOld )
      ELSE
        Avalue = ' Undefined'
      ENDIF
      Return
      END
      SUBROUTINE SetChr(String,Value)
C-----------------------------------------------------------------------------------------------------------------------
C This routine translates a value into a string of characters. If the ABS(value) is less than Apprx0, the string is set
C to blanks.
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE 'Hydrazin.CMB'
      CHaracter String*12
      IF (ABS(Value).GT.Apprx0) THEN
         Write(String,1024) Value
 1024     Format(G12.5)
      ELSE
         String = '     0      '
      ENDIF
      Return
      END
      Logical Function NotEqu(Vabove,Value,Vbelow)
C-----------------------------------------------------------------------------------------------------------------------
C This function determines whether the value is different from the flankers
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE 'Hydrazin.CMB'
      Character AbvStr*12,ValStr*12,BelStr*12
      Write(AbvStr,1000) Vabove
      Write(ValStr,1000) Value
      Write(BelStr,1000) Vbelow
 1000   Format(G12.5)
      If (ValStr.EQ.AbvStr.AND.ValStr.EQ.BelStr) THEN
         NotEqu = .false.
      ELSE
         NotEqu = .true.
      ENDIF
      Return
      END
      Subroutine RepItr
C-----------------------------------------------------------------------------------------------------------------------
C This routine reports the iterative values
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE 'Hydrazin.CMB'
      Last = 1
      Do 10 I = 1,ItrLim
   10   If (ItrTrk(I).GT.0.0) Last = I
      Write(12,1000) 'Iteration Trace:'
 1000   Format(//,A)
      Write(12,1010)('  <',I,':',100.0*ItrTrk(I)/Irepet,'%>',I=1,Last)
 1010   Format(6(A,I2,A,G12.5,A))
      Return
```

```
      END
      Double Precision Function BdMean(Ion,Idep)
C-----------------------------------------------------------------------------------------------------------------------
C This routine calculates n +1/2 value for the bound phase
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE 'Hydrazin.CMB'
      BdMean = 0.5*(Bound(Ion,Idep) + OldBnd(Ion,Idep))
      Return
      END
      Double Precision Function FxMean(Ion,Idep)
C-----------------------------------------------------------------------------------------------------------------------
C This routine calculates n +1/2 value for the tightly adsorbed phase
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE 'Hydrazin.CMB'
      FxMean = 0.5*(FixSrb(Ion,Idep) + OldFix(Ion,Idep))
      Return
      END
      Double Precision Function CdMean(Ion,Idep)
C-----------------------------------------------------------------------------------------------------------------------
C This routine calculates n +1/2 value for the dissolved phase
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE 'Hydrazin.CMB'
      CdMean = 0.5*(Cdissv(Ion,Idep) + OldDis(Ion,Idep))
      Return
      END
      Double Precision Function AdMean(Ion,Idep)
C-----------------------------------------------------------------------------------------------------------------------
C This routine calculates n +1/2 value for the adsorbed phase
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE 'Hydrazin.CMB'
      AdMean = 0.5*(Adsorb(Ion,Idep) + OldAds(Ion,Idep))
      Return
      END
      SUBROUTINE Mbalck
C-----------------------------------------------------------------------------------------------------------------------
C This routine performs a Mass Balance check
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE 'Hydrazin.CMB'
      CALL Mbalan
      DO 10 Ion = 1,NinUse                                          {for each ion,total up the amount of material
        ERRors(Ion,1) = 0.0D+00
        If (UseIon(Ion)) THEN
          CACT(Ion) = CINIT(Ion) + FLOWIN(Ion)                      {the amount that should be in the system}
          CPRED(Ion,1) = TotNow(Ion,1)+TotNow(Ion,2)+TotNow(Ion,3)+
     &      TotNow(Ion,4)+ TotNow(Ion,5)+ FLOWOU(Ion) + SrbOut(Ion)
          IF (CACT(Ion).GT.Apprx0) THEN
            Errors(Ion,1) = 100.0*(CPRED(Ion,1)-CACT(Ion))/CACT(Ion)  {the error between the two figures}
          ENDIF
        ENDIF
   10 CONTINUE
      RETURN
      END
      SUBROUTINE MBalan
C-----------------------------------------------------------------------------------------------------------------------
C This routine performs the mass balance calculation by summing up the contents of the column in the 2 phases
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE 'Hydrazin.CMB'
      BALSOL = 0.0D+00
      BALSOR = 0.0D+00
      DO 60 Ion = 1,NinUse
        DO 10 Itype = 1,5
   10     TotNow(Ion,Itype) = 0.0D+00
        DO 20 Idep = 2,Nodes-1
          If (Ion.LE.2) THEN
            TotNow(Ion,3) = TotNow(Ion,3) + Adsorb(Ion,Idep)
            TotNow(Ion,4) = TotNow(Ion,4) + FixSrb(Ion,Idep)
            TotNow(Ion,5) = TotNow(Ion,5) + Bound(Ion,Idep)
          ENDIF
          TotNow(Ion,2) = TotNow(Ion,2) + Csorb(Ion,Idep)
   20     TotNow(Ion,1) = TotNow(Ion,1) + Cdissv(Ion,Idep)          {and that in solution phase corrected for theta}
        TotNow(Ion,1) = TotNow(Ion,1) * ThetaS * Dx                 {scale the solution total by the nodal size}
        DO 25 Itype = 2,5
          TotNow(Ion,Itype) = TotNow(Ion,Itype) * BULKDN * DX       {and the sorbed side by bulk density & nodal size}
   25     BalSor = BalSor + TotNow(Ion,Itype)
   60   BALSOL = BALSOL + TotNow(Ion,1)                              {then toal across the ions}
      RETURN
      END
      SUBROUTINE RepIon
C-----------------------------------------------------------------------------------------------------------------------
C This routine reports the condition of the column's ionic state whenever requested.
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE 'Hydrazin.CMB'
      Dimension Comann(MaxIon),Camob(MaxNod),TOTINT(MaxIon,5),
     &          StuffI(MaxIon),CaTotI(MaxNod),PrtNow(MaxIon,2)
      Logical NotEqu,PrtVal
      CHARACTER*15 Commnt,Avalue(5)*12,Blanks*12
      DATA Blanks/'    '/
      IF (Irepet.LE.1) THEN                                         {If starting ...}
         CALL Mbalck                                                {scan the column to establish initial conditions for
         ORGSOL = BalSOL                                            { mass balance calculations}
         ORGSOR = BalSor
         DO 1 Ion = 1,Nions
           Cinit(Ion) = 0.0D+00
           Do 1 Itype = 1,5
             CINIT(Ion) = Cinit(Ion) + TotNow(Ion,Itype)            {such as initial concentrations for all ions}
    1        TOTINT(Ion,Itype) = TotNow(Ion,Itype)
      ENDIF
      Hour = Time/3600.0                                            {print a header block for the report}
      Day = Hour/24.0
      WRITE(12,1020)CHar(12),RLABEL,TIME,HOUR,Day,DX,DT,Nodes,POREVL,
     &              IREPET
```

```
1020  FORMAT(A,'-',57(1H-),'<SOLUTE REPORT>',57('-'),'-',/,24X,A, ,
     &  '  Time = ',F10.2,' Sec. = ',F6.1,' Hrs.= ',F6.1,' Days: ',
     &  'Dx = ',F5.2,'; Dt = ',F7.2,'; Nodes = ',I3,
     &  ' for following block',/,
     &  '  cumulative pore volume up to Now = ',F12.1,49X,
     &  ' Total Iterations=',I9)
      CUMCTR = 0.0D+00                                          {Perform the CEC error analysis}
      DO  3 I = 2,Nodes-1
          DO 3 J = 2,3
    3       CUMCTR = CUMCTR + Csorb(J,I)
          CumCtr = CumCtr/Float(Nodes-2)
          Call FilVal(Avalue(1),CumCtr,CECfix)
      WRITE(12,1025) CUMCTR,CECFix,Avalue(1)
1025  FORMAT('  Average CEC in column= ',G12.5,' Specified CEC=',
     &       G12.5,' Error%=',A)
      If (Irepet.GT.0) Call Mbalck                              {then the mass balance analysis}
      TINPUT = 0.0D+00
      TOUTPT = 0.0D+00
      DO 4 Ion = 1,Nions
         TINPUT = TINPUT + FLOWIN(Ion)
    4    TOUTPT = TOUTPT + FLOWOU(Ion)
      ORGEST = TINPUT + ORGSOL + ORGSOR                         {best value for input + orginial amounts}
      BESTES = (TINPUT-OLDINP) + OLDSOL + OLDSRb                {best estimate for errors since last report}
      CUMCON = BALSOL + BALSOR + TOUTPT                         {comparison value for overall error}
      CURCON = BALSOL + BALSOR + (TOUTPT-OLDOUT)                {comparison value for errors since last report}
      StuffT = 0.0
      DO 50 Ion = 1,Nions                                       {for all ions that are active in the simulation...}
        If (UseIon(Ion)) THEN
          WRITE(12,1030) SOLUTE(Ion)
1030        FORMAT(/,32X,'---> Following summary ',
     &        'applies to Ion: ',A,' <---',/)
          StuffI(Ion) = DABS(Cpred(Ion,1) - Cact(Ion))            {acquire discrepancy values and error values}
          StuffT      = StuffT + StuffI(Ion)
          Call SetChr(Avalue(1),  100.0*FLowOu(Ion)/
     &                          (Cinit(Ion)+Flowin(Ion)) )
          WRITE(12,1040) FLOWIN(Ion),FLOWOU(Ion),CINIT(Ion),    {then report the results}
     &       TotNow(Ion,1)+TotNow(Ion,2),TotNow(Ion,3),TotNow(Ion,4),
     &       Avalue(1)
1040      FORMAT('  Total Flow in = ',G10.3,' Total Flow out = ',
     &       G10.3,' Initial storage = ',G10.3,' Present storage= ',
     &       G10.3,/,' ',38X,'Dissolved total = ',G10.3,
     &       ' Sorbed total = ',G10.3,/,' Percent left from system (',
     &       'Out*100/ Initial+Input) = ',A)
          If (Ion.LE.2) THEN
            Call FilVal(Avalue(1),SrbOut(Ion),FlowIn(Ion))
            Write(12,2040)SrbOut(Ion),Avalue(1),Hydrla(Ion),HydrIn(Ion)
2040        Format(' Irreversible sorption accounted for: ',G12.5,
     &       ' % of Input adsorbed=',A,/,' Hydrolysis: Total=',G12.5,
     &       ' Input=',G12.5)
            If (Sponge.GT.0.0) THEN
              AmLeft=100.0*TotNow(Ion,5)/(Bulkdn*Dx*Sponge*(Nodes-2))
            ELSE
              AmLeft = 0.0
            ENDIF
            Write(12,2045) TotNow(Ion,3),TotNow(Ion,4),TotNow(Ion,5),
     &                     AmLeft
2045         Format(' Total: Physi-sorbed=',G12.5,' chemi-sorbed=',
     &          G12.5,' Bound=',G12.5,' %=',G12.5)
          ENDIF
          Call FilVal(Avalue(1),Cpred(Ion,1),Cact(Ion))
          If (Irepet.LE.1) Avalue(1) = ' Undefined'
          WRITE(12,1050) CACT(Ion),CPRED(Ion,1),Avalue(1)
1050       FORMAT(' Sum of Input & initial conc. = ',g10.3,' Sum of ',
     &       'stored & outflow = ',G10.3,' Error%= -----<',A,'>-----')
        ENDIF
   50 CONTINUE
      If (UseIon(1).AND.UseIon(2)) THEN
        WRITE(12,1030) 'Hydrazine + Hydrazinium'
        Call SetChr(Avalue(1),100.0*(FlowOu(1)+FLowOu(2))/
     &                    (Cinit(1)+Flowin(1)+Cinit(2)+Flowin(2)) )
        WRITE(12,1040)FLOWIN(1)+FlowIn(2),FLOWOU(1)+FlowOu(2),CINIT(1)+
     &    Cinit(2),TotNow(1,1)+TotNow(1,2)+TotNow(2,1)+TotNow(2,2),
     &    TotNow(1,1)+TotNow(2,1),TotNow(1,2)+TotNow(2,2),Avalue(1)
        Call FilVal(Avalue(1),SrbOut(1)+SrbOut(2),FlowIn(1)+FlowIn(2))
        Write(12,2040) SrbOut(1)+SrbOut(2),Avalue(1),
     &                 Hydrla(1)+HydrLa(2),HydrIn(1)+HydrIn(2)
        If (Sponge.GT.0.0) THEN
          AmLeft = 100.0*(TotNow(1,5)+TotNow(2,5))/
     &                      (Bulkdn*Dx*Sponge*(Nodes-2))
        ELSE
          AmLeft = 0.0
        ENDIF
        Write(12,2045)TotNow(1,3)+TotNow(2,3),TotNow(1,4)+TotNow(2,4),
     &                TotNow(1,5)+TotNow(2,5),AmLeft
        Call FilVal(Avalue(1),Cpred(1,1)+Cpred(2,1),Cact(1)+Cact(2))
        If (Irepet.LE.1) Avalue(1) = ' Undefined'
        WRITE(12,1050) CACT(1)+Cact(2),CPRED(1,1)+Cpred(2,1),Avalue(1)
      ENDIF
      Do 52 Ion = 1,Nions
        If (UseIon(Ion)) THEN                                   {present the discrepancy analysis}
          Call FilVal(Avalue(1),StuffI(Ion)+StuffT,StuffT)
          Write(12,2050) Solute(Ion),StuffI(Ion),StuffT,Avalue(1)
2050        Format(' ',A,' Mass Discrepancy =',G12.5,
     &       ' of total discrepancy=',G12.5,' Error=',A,' % ')
        ENDIF
   52 Continue
      WRITE(12,2051)
2051      FORMAT(/,33X,'---> Following summary ',               {then the overall error report}
     &       'applies to all Ions taken together <---',/)
      WRITE(12,1052) TINPUT,OLDINP,TOUTPT,OLDOUT,BALSOL,OLDSOL,
     &               BALSOR,OLDSRB,SrbOut(1)+SrbOut(2),OldSr
```

```
1052  FORMAT('  Total current input=',G12.5,' Old input=',G12.5,
     &     ' Current output=',G12.5,' Old output=',G12.5,/,
     &     '  Total current dissolved storage=',G12.5,' old=',g12.5,
     &     ' current sorbed storage=',g12.5,' old=',G12.5,/,
     &     '   Total decayed=',G12.5,' Old decayed=',G12.5)
      Call FilVal(Avalue(1),CurCon+SrbOut(1)+SrbOut(2)-OldDk,BestEs)
      WRITE(12,1053) Avalue(1)
1053      FORMAT(45X,'Error since last report = ',A)
      Call FilVal(Avalue(1),CumCon+SrbOut(1)+SrbOut(2),Orgest)
      WRITE(12,1055) Avalue(1)
1055      FORMAT(45X,'Error since initiation  = ',A)
      OLDINP = TINPUT
      OLDOUT = TOUTPT
      OLDSOL = BALSOL
      OLDSRB = BALSOR
      OldDk  = SrbOut(1) + SrbOut(2)
      WRITE(12,1060)
1060  FORMAT('+',129(1H-),'+',/)                                    (now show am image of the column)
      DO 110 Ion = 1,Nions
        If (.Not.UseIon(Ion)) Goto 110
         WRITE(12,1100) '*'
1100    Format(' Ion number & Designation  Depth     Time    Pore ',
     &    'Volume   Dissolved   Adsorbed      Sorbed     Total (/g)',
     &    /,A)
        Lines = 0
        Do 99 I = 3,Nodes-2
  99      If (NotEqu(Cdissv(Ion,I-1),Cdissv(Ion,I),
     &                   Cdissv(Ion,I+1))) Lines = Lines + 1
        LinOut = MaxDiv(Lines,50)
        If (LinOut.LE.0) LinOut = 1
        DO 100 I = 1,Nodes
          If (I.LE.2.OR.I.GE.Nodes-2) THEN                          (always print the top & bottom & any node that is
             PrtVal = .true.                                        (different from its neighbors)
          ELSE
             PrtVal = NotEqu(Cdissv(Ion,I-1),Cdissv(Ion,I),
     &                           Cdissv(Ion,I+1))
             If (MOD(I,LinOut).NE.0) PrtVal = .false.
          ENDIF
          If (PrtVal) THEN
            DEPTH = DX *(I-1.5)
            COMMNT = '              '
            IF (I.EQ.1.OR.I.EQ.Nodes) COMMNT = ' Not in Column'
            Call SetChr(Avalue(1),Cdissv(Ion,I))
            Call SetChr(Avalue(2),Adsorb(Ion,I))
            If (Ion.GE.2.AND.Ion.LE.3) THEN
              Call SetChr(Avalue(3),Csorb(Ion,I))
              Call SetChr(Avalue(4),ThetaS*Cdissv(Ion,I)/
     &                       BulkDn + Csorb(Ion,I))
            ELSE
              Avalue(3) = ' '
              Call SetChr(Avalue(4),ThetaS*Cdissv(Ion,I)/BulkDn)
            ENDIF
            WRITE(12,1110)Ion,SOLUTE(Ion),DEPTH,TIME,
     &                    POREVL,(Avalue(Ival),Ival=1,4),COMMNT
1110          FORMAT(' ',I1,' : ',A,4X,F7.3,F9.1,G15.5,5A)
          ENDIF
 100    CONTINUE
        WRITE(12,1115) '*'
 110  Continue
1115      FORMAT(A)
      WRITE(12,1100) '*'
      DO 101 IDEP = 1,Nodes                                         (then do the same for the totalled column:
         CAMOB(IDEP) = 0.0D+00
         CaTot1(Idep) = 0.0D+00
         DO 101 Ion = 2,3
           CaTot1(Idep) = CaTot1(Idep) +
     &       ThetaS*Cdissv(Ion,Idep)/BulkDn + Csorb(Ion,Idep)
 101       CAMOB(IDEP) = CAMOB(IDEP) + Cdissv(Ion,IDEP)
      Avalue(3) = ' '
      DO 105 I = 1,Nodes
        If (I.LE.2.OR.I.GE.Nodes-2) THEN
          PrtVal = .true.
        ELSE
          PrtVal = NotEqu(Camob(I-1),Camob(I),Camob(I+1))
          If (MOD(I,LinOut).NE.0) PrtVal = .false.
        ENDIF
        If (PrtVal) THEN
          DEPTH = DX *(I-1.5)
          COMMNT = '              '
          IF (I.EQ.1.OR.I.EQ.Nodes) COMMNT = ' Not in Column'
          Call SetChr(Avalue(1),Camob(I))
          Call SetChr(Avalue(4),CaTot1(I))
          WRITE(12,1110)Ion,'Common Anion        ',DEPTH,TIME,
     &                  POREVL,(Avalue(J),J=1,4),COMMNT
        ENDIF
 105  Continue
      WRITE(12,1115) '*'
      Itmptr = Itmptr + 1
      RETURN
      END
      Subroutine ChkFlx
C-------------------------------------------------------------------------------------------------------------------
C This routine handles setting the flux & velocity if they should change during as run
C-------------------------------------------------------------------------------------------------------------------
      INCLUDE 'Hydrazin.CMB'
      IF ((IflxPt.LE.0).OR.                                         (if this is the first time this routine is called or
     &    (Time.GE.FluxTm(IflxPt+1).AND.(IflxPt+1).LE.NFLUX)) THEN  (it is time to reset the flux values...)
        IflxPt = IflxPt + 1                                         (increment the flux ptr and reset the flux & veloc.)
        DtStar = .true.
        LckDwn = 0
        Write(12,*) ' '
```

```
      Write(12,'(I3)') IflxPt,PoreVl,
     &   (Solute(J)(1:6),FluxAm(IflxPt,J),J=0,NinUse)
 1010   Format(26X,'Flux(',I2,') at ',F6.3,' pvi to : ', ,
     &                  4(1X,A,'=',G12.5))
       ENDIF
       LokDwn = LokDwn + 1
       Return
       END
       Subroutine DtChan
C-------------------------------------------------------------------------------------------------------------------------
C This routine resets the Dt
C-------------------------------------------------------------------------------------------------------------------------
       INCLUDE 'Hydrazin.CMB'
       If (LokDwn.LE.10) THEN
         Dt = Dtmin
       ELSE
         DtNow = Dt
         If (IflxPt.GT.0.AND.Iflxpt.LT.Nflux) THEN
           Nsteps = NINT(-1.0 + (FluxTm(Iflxpt+1)-(Time+Dt))/
     &                    (Dt/2.0))
           If (Nsteps.LT.0) Nsteps = 0
           If (Nsteps.LT.20) THEN
             Dt = Dt*Nsteps*DtStep + DtMin
             DtDown = .true.
           ENDIF
         ENDIF
         If (DtStar.AND.LokDwn.GT.10) THEN                    {either true initially or when there is a flux change}
           DtStar = .false.
           DtAltr = .true.                                    {prevent entering here until next flux change, ready}
           DtDown = .false.
           Dt = Dtmin + DtStep*DtMax                          {error and set Dt to the minimum again}
         ENDIF
         IF (DTALTR.OR.DtDown) THEN
           CALL Mbalck
           If (.not.DtDown) DT = DT + DtStep*DtMax            { recalculate the equation constants used in the
           If (Dt.LE.DtMin+DtStep*DtMax) Dt = DtMin+DtStep*DtMax
           IF (DT.GE.DTMax) THEN                              {up to the maximum limit set by the parameter file}
             DT = DTMax
             DTALTR = .FALSE.
           ENDIF
         ENDIF
       ENDIF
       Return
       END
       Subroutine RepPrf
C-------------------------------------------------------------------------------------------------------------------------
C This routine reports a profile for all active ions
C-------------------------------------------------------------------------------------------------------------------------
       INCLUDE 'Hydrazin.CMB'
       Character Label*8,Rest*65,StrVal*7
       If (Irepet.GT.1) THEN
         Write(StrVal,1000) PoreVl
 1000     Format(F7.4)
         Rest = StrVal//'-'//Rlabel(1:57)                   {setup a label-continuation text-string}
         Do 30 Ion = 1,Nions
           If (USeIon(Ion)) THEN                            {for all active ions}
             Label = Solute(Ion)(1:6)//'Cd'
             Write(14,1010) Label,Nodes-2,Rest
 1010          Format(A,I5,1X,A)
             Do 5 Idep = 2,Nodes-1
    5          Write(14,1020) Dx*(Idep-1.5),Cdissv(Ion,Idep)  {write the depth and the concentration}
 1020          Format(2G12.5)
             If (CEC.GT.Apprx0) THEN                        {then the sorbed concentrations (if CEC >0)
               Label(8:8) = 'X'
               Write(14,1010) Label,Nodes-2,Rest
               Do 10 Idep = 2,Nodes-1
   10            Write(14,1020) Dx*(Idep-1.5),Csorb(Ion,Idep)
             ENDIF
             If (Ion.LE.2) THEN
               If (RatRev(Ion,1)*RatRev(Ion,2).GT.Apprx0) THEN  {if there is a loosely sorbed compartment...}
                 Label(8:8) = 'P'                             {report on it}
                 Write(14,1010) Label,Nodes-2,Rest
                 Do 15 Idep = 2,Nodes-1
   15              Write(14,1020) Dx*(Idep-1.5),Adsorb(Ion,Idep)
               ENDIF
               If (RatRev(Ion,4)*RatRev(Ion,5).GT.Apprx0) THEN  {if there is a tightly sorbed compartment...}
                 Label(8:8) = 'C'                             {report on it}
                 Write(14,1010) Label,Nodes-2,Rest
                 Do 20 Idep = 2,Nodes-1
   20              Write(14,1020) Dx*(Idep-1.5),Fixsrb(Ion,Idep)
               ENDIF
               If (RatRev(Ion,6)*RatRev(Ion,7).GT.Apprx0) THEN  {if there is a permanently bound compartment...}
                 Label(8:8) = 'B'                             {report on it}
                 Write(14,1010) Label,Nodes-2,Rest
                 Do 25 Idep = 2,Nodes-1
   25              Write(14,1020) Dx*(Idep-1.5),Bound(Ion,Idep)
               ENDIF
             ENDIF
           ENDIF
   30    Continue
       ENDIF
       Return
       END
       Subroutine EndPgm(Text)
C-------------------------------------------------------------------------------------------------------------------------
C This routine terminates the program with an error message
C-------------------------------------------------------------------------------------------------------------------------
       INCLUDE 'Hydrazin.CMB'
       Logical Stopit
       Character*(*) Text
       Data Stopit/.true./
```

```
      Write(*,1000) Text
 1000   Format(A)
      If (Stopit) Stop
      Return
      END
      Subroutine StrCap(String)
C-----------------------------------------------------------------------------------------------------------------------
C        Capitialize the String
C-----------------------------------------------------------------------------------------------------------------------
      CHaracter*(*) String
      Call StrLen(String,Length)
      IF (Length.GT.0) THEN
        Do 10 I = 1,Length
   10     If (String(I:I).GE.'a'.AND.String(I:I).LE.'z') String(I:I) =
     &          Char(Ichar('A') + Ichar(String(I:I))-Ichar('a'))
      ENDIF
      Return
      END
      SUBROUTINE STRLEN(STRING,Length)
C-----------------------------------------------------------------------------------------------------------------------
C        returns as LENGTH the position of the last non-blank char. in STRING
C-----------------------------------------------------------------------------------------------------------------------
      CHARACTER*(*) STRING
      LENGTH = 0
      LstChr = Len(String)
      DO 10 I = 1,LstChr
         J = LstChr + 1 - I
         IF (STRING(J:J).NE. ' ') THEN
            Length = J
            I = LstChr
         ENDIF
   10 CONTINUE
      RETURN
      END
      Subroutine WritDP(Label,Rest,Nobs,X,Y,IO)
C-----------------------------------------------------------------------------------------------------------------------
C This routine writes a plot data-line to an already open file
C-----------------------------------------------------------------------------------------------------------------------
      Implicit Double precision (A-H,O-Z)
      Dimension X(*),Y(*)
      Character Label*8,Rest*65
      Write(IO,1000) Label,Nobs,Rest(1:LenOrl(Rest))
 1000   Format(A,I5,1X,A)
      Write(IO,1010) (X(I),Y(I),I=1,Nobs)
 1010   Format(2G12.5)
      Return
      END
      Integer Function Iextnt(FilNam)
C-----------------------------------------------------------------------------------------------------------------------
C This routine returns the start position of a filename extension
C-----------------------------------------------------------------------------------------------------------------------
      Character FilNam*15
      Istop = 0
      DO 130 I = 1,Len(FilNam)
         IF (FilNam(I:I).EQ.'.'.OR.FilNam(I:I).EQ.' ') THEN
            Istop = I
            Goto 140
         ENDIF
  130 Continue
  140 Iextnt = Istop
      Return
      END
      Subroutine EchoLo(Bolean)
C-----------------------------------------------------------------------------------------------------------------------
C This routine reads and echos a boolean setting from the parameter file
C-----------------------------------------------------------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      Character ALine*80,YesNo*3
      Logical Bolean
      Read(10,1000) YesNo,ALine
 1000   Format(12X,2A)
        Write(12,1000) YesNo,ALine(1:LenOrl(ALine))
        Bolean = Str2Lo(YesNo)
      Return
      END
      Subroutine EchoRl(Avalue)
C-----------------------------------------------------------------------------------------------------------------------
C This routine reads and echos a real setting from the parameter file
C-----------------------------------------------------------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      Character ALine*80
      Read(10,1000) Avalue,ALine
 1000   Format(G15.7,A)
        Write(12,1000) Avalue,ALine(1:LenOrl(ALine))
      Return
      END
      Subroutine EchoIn(Ivalue)
C-----------------------------------------------------------------------------------------------------------------------
C This routine reads and echos an integer setting from the parameter file
C-----------------------------------------------------------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      Character ALine*80
      Read(10,1000) Ivalue,ALine
 1000   Format(I15,A)
        Write(12,1000) Ivalue,ALine(1:LenOrl(ALine))
      Return
      END
      Subroutine EchoSt(Text)
C-----------------------------------------------------------------------------------------------------------------------
C This routine reads and echos a string setting from the parameter file
C-----------------------------------------------------------------------------------------------------------------------
```

```
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      Character*(*) Text,Aline*80
      LenTxt = Len(Text)
      Read(10,1000) Aline
 1000  Format(A)
        Write(12,1000) Aline(1:LenOrl(Aline))
        Text = Aline(1:LenTxt)
      Return
      END
      Integer Function LenOrl(STRING)
C-----------------------------------------------------------------------------------------------------------------
C        returns as LENGTH the position of the last non-blank char. in STRING
C-----------------------------------------------------------------------------------------------------------------
      CHARACTER*(*) STRING
      LENGTH = 0
      LstChr = Len(String)
      DO 10 I = 1,LstChr
         J = LstChr + 1 - I
         IF (STRING(J:J).NE. ' ') THEN
            Length = J
            Goto 20
         ENDIF
   10 CONTINUE
   20 If (Length.LE.0) Length = 1
      LenOrl = Length
      RETURN
      END
      Logical Function Str2Lo(Text)
C-----------------------------------------------------------------------------------------------------------------
C this routine converts a yes/no answer into a logical value
C-----------------------------------------------------------------------------------------------------------------
      Character*(*) Text
      Call StrCap(Text)
      Call ShfTxt(Text)
      If (Text(1:1).EQ.'Y') THEN
        Str2Lo = .true.
      ELSE
        Str2Lo = .false.
      ENDIF
      Return
      END
```

D. Example of a Parameter files for the Hydrazine program

```
E2KFstCL.RES     : Filename for results storage
11.9ppm-5 cm/hr cont-Kin- (BTC 24)
 30.0          :Column Length (cm)
32400.0         :Time to terminate simulation (sec)
 .25           :Saturated Water Content
 .13889E-02     :Saturated conductivity
 1.63          :Bulk Density (gm/cc)
 .2694E-03      :Dispersion intercept
0.1105E-02      :Bound Maximum capacity
          No :---> Activate Ion-Exchange <---
         Yes :---> Activate Reversible Sorption <---
         Yes :Activate Kinetic Reversible Sorption
 0.0           :Hydrazine Rate Coefficient (kf)
 0.0           :Hydrazine Rate Coefficient (kb)
 1.0           :Hydrazine exponent Coefficient (N)
 0.0           :Hydrazine Rate Coefficient (kff)
 0.0           :Hydrazine Rate Coefficient (kbb)
 0.0           :Hydrazine Rate Coefficient (kp)
 0.0           :Hydrazine Rate Coefficient (kq)
         Yes :Activate Kinetic Reversible Sorption
 0.100E-02      :Hydrazinium Rate Coefficient (kf)
 0.500E-02      :Hydrazinium Rate Coefficient (kb)
 1.00          :Hydrazinium Exponent Coefficient (N)
 0.300E-03      :Hydrazinium Rate Coefficient (kff)
 0.700E-03      :Hydrazinium Rate Coefficient (kbb)
 0.0           :Hydrazinium Rate Coefficient (kp)
 0.100E-02      :Hydrazinium Rate Coefficient (kq)
          No :---> Activate Irreversible Sorption <---
          No :---> Activate Microbial Degradation <---
          No :---> Activate Auto-Oxidation <---
          No :---> Activate Hydrolysis Conversion <---
           1 :Number of boundary condition specs
 0.0           :----> Depth (cm) <----
 .0            :Hydrazine in solution (meq/cc)
 .1E-09         :Hydrazinium in solution (meq/cc)
 .0            :Hydrazinium sorbed (meq/g)
 .01           :Calcium in solution (meq/cc)
 .0            :Calcium sorbed (meq/g)
 .0            :Oxygen in solution (meq/cc)
 .0            :Cation Exchange Capacity (meq/g)
 .25           :Dx = Nodal increment (cm)
  50.          :Maximum Dt (secs)
           1 :Number of flux changes during run
 0.0           :Time (sec) of flux change
 5.0           :Flux (cm/hr) imposed
 .0            :Hydrazinium in solution (meq/cc)
 .23771E-03     :Hydrazine in solution (meq/cc)
 .01           :Calcium in solution (meq/cc)
 .0            :Oxygen in solution (meq/cc)
           1 :Number of reports to be generated
32400.0         :Report to be generated (sec)
 1.0           :Divisor scale factor for breakthrough
 .23771E-03     :Divisor scale factor for breakthrough
 .01           :Divisor scale factor for breakthrough
 1.0           :Divisor scale factor for breakthrough
          No :Feedback for debugging
         100 :abortion check periodicity
 0.10          :Tolerance Limit in percentage
          20 :Number of passes before iterative failure
 .50000000E-01 :Increment for Dt - fractional
-30.0          :Power for approximately zero value
         Yes :Activate the two-site model
```

APPENDIX C

COMPUTER CODE FOR TWO-DIMENSIONAL MODEL

2. Computer Code for Two-Dimensional Model

Program for Creating Two-Dimensional Boundary Conditions File (MHD)

```
      Program SysMak
C-------------------------------------------------------------------------------
      Dimension Head(41,41),Solute(41,41),Surfac(41),Seep(41),
     &          Drain(41)
      Integer Surfac,Seep,Drain,Row,Col
      Character Title*80,Finput*15
      Logical AskQus,DranOn,SeepOn,SurfOn,AltHed,AltSol
      Call Page('Create 2-D MHD file')                    {Initialize paging routines}
    1 Call Genral(Title,Dx,Finput,Ncols,Nrows,Hval,Sval,  {get general description of system}
     &                    DranOn,SeepOn,SurfOn,AltHed,AltSol)
        DO 30 I = 1,Nrows                                 {set seep & drain to non-existence}
          Seep(I) = 1
          Drain(I) = 1
          Do 30 J = 1,Ncols                               {and matrices to uniform values}
            Solute(J,I) = Sval
   30       Head(J,I) = Hval
        Do 35 J = 1,Ncols                                 {and the surface to a single zone}
   35     Surfac(J) = 1
        If (SeepOn) Call SeepDf(Seep,Nrows)               {define the seep}
        If (DranOn) Call DranDf(Drain,Nrows)              {define the drain}
        If (SurfOn) Call SurfDf(Surfac,Ncols)             {define the surface zones}
        If (AltHed) Call Alter('Head  ',Head,Ncols,Nrows) {set up special head regions}
        If (AltSol) Call Alter('Solute',Solute,Ncols,Nrows) {set up special solute regions}
        Call PrtDat(Title,Dx,Ncols,Nrows,Surfac,Seep,Drain, {create the MHD file}
     &                          Head,Solute)
      If (AskQus('Rerun program?',.false.,22)) Goto 1
      END
      Subroutine Genral(Title,Dx,Finput,Ncols,Nrows,Hval,Sval,DranOn,
     &                          SeepOn,SurfOn,AltHed,AltSol)
C-------------------------------------------------------------------------------
C-------------------------------------------------------------------------------
      Save
      Dimension Answer(12),Query(12),String(12)
      Character Title*80,Finput*15,String*80,Query*80
      Logical AskQus,DranOn,SeepOn,SurfOn,AltHed,AltSol,BadVal,
     &        Rel2Lo,Error
      Real Lo2Rel
      Data Nitems/12/,LenQry/50/,Query/
     & '0501S? Title',
     & '0602S? File name',
     & '0803I? Number of columns (3..40)',
     & '0904I? Number of rows (3..40)',
     & '1005R? Overall head value',
     & '1106R? Overall solute value',
     & '1307LD A seep zone exists',
     & '1408LD A drain zone exists',
     & '1509LD Multiple surface zones exist',
     & '1710LD Head subregions exist',
     & '1811LD Solute subregions exist',
     & '2012R? Node spacing (cm) (=Dx)'/
      Data Answer/65.,15.,4*0.,5*2.,0./
    1 BadVal = .false.
      Call MenFil(LenQry,Nitems,Query,Answer,String,.true.)
      Title = String(1)
        Call StrLen(Title,Length)
        If (Length.LE.1) Title = 'No title given'
      Finput = String(2)
        Call StrLen(Finput,Length)
        If (Length.LE.0) THEN
          BadVal = .true.
        ELSE
          Call OpnFil(Finput,12,'NEW',Error)
          If (.Not.Error) Goto 10
          BadVal = .true.
        ENDIF
   10   If (BadVal) THEN
          String(2) = 'BAD NAME!'
          Goto 1
        ENDIF
        Ncols = NINT(Answer(3))
        If (Ncols.LE.3.OR.Ncols.GT.40) THEN
          Query(3)(6:6) = '?'
          Goto 1
        ENDIF
        Nrows = NINT(Answer(4))
        If (Nrows.LE.3.OR.Nrows.GT.40) THEN
          Query(4)(6:6) = '?'
          Goto 1
        ENDIF
      Hval = Answer(5)
      Sval = Answer(6)
      Seepon = Rel2Lo(Answer(7))
      DranOn = Rel2Lo(Answer(8))
      SurfOn = Rel2Lo(ANswer(9))
      AltHed = Rel2Lo(Answer(10))
      AltSol = Rel2Lo(Answer(11))
      Dx     = Answer(12)
      If (Dx.LE.0.0) THEN
```

```
        Query(12)(6:6) = '?'
        Goto 1
      ENDIF
      Return
      END
      Subroutine SeepDf(Seep,Nrows)
C-------------------------------------------------------------------------------
C-------------------------------------------------------------------------------
      Dimension Seep(41),Answer(3),Query(3),String(3)
      Character Query*80,String*80
      Integer Seep
      Data Nitems/3/,LenQry/60/,Query/
     & '0700CC Definition of a seep (left side of system)',
     & '1001I? First row included in the seep (>1)',
     & '1202I? Last row included in the seep (<xx)'/
      Write(Query(3)(40:41),1000) Nrows
 1000   Format(I2)
    1 Call MenFil(LenQry,Nitems,Query,Answer,String,.true.)
      If (NINT(Answer(2)).LT.2.OR.NINT(Answer(2)).GT.Nrows-1.OR.
     &      Nint(Answer(2)).GT.NINT(Answer(3))) THEN
        Query(2)(6:6) = '?'
        Goto 1
      ENDIF
      If (NINT(Answer(3)).LT.2.OR.NINT(Answer(3)).GT.Nrows-1) THEN
        Query(3)(6:6) = '?'
        Goto 1
      ENDIF
      Do 10 I = NINT(Answer(2)),Nint(Answer(3))
   10   Seep(I) = 0
      Return
      END
      Subroutine DranDf(Drain,Nrows)
C-------------------------------------------------------------------------------
C-------------------------------------------------------------------------------
      Dimension Drain(41),Answer(3),Query(3),String(3)
      Character Query*80,String*80
      Integer Drain
      Data Nitems/3/,LenQry/60/,Query/
     & '0700CC Definition of a Drain (right side of system)',
     & '1001I? First row included in the drain (>1)',
     & '1202I? Last row included in the drain (<xx)'/
      Write(Query(3)(41:42),1000) Nrows
 1000   Format(I2)
    1 Call MenFil(LenQry,Nitems,Query,ANswer,String,.true.)
      If (NINT(Answer(2)).LT.2.OR.NINT(Answer(2)).GT.Nrows-1.OR.
     &      Nint(Answer(2)).GT.NINT(Answer(3))) THEN
        Query(2)(6:6) = '?'
        Goto 1
      ENDIF
      If (NINT(Answer(3)).LT.2.OR.NINT(Answer(3)).GT.Nrows-1) THEN
        Query(3)(6:6) = '?'
        Goto 1
      ENDIF
      Do 10 I = NINT(Answer(2)),Nint(Answer(3))
   10   Drain(I) = 0
      Return
      END
      Subroutine SurfDf(Surfac,Ncols)
C-------------------------------------------------------------------------------
C-------------------------------------------------------------------------------
      Dimension Surfac(41),Test(41)
      Character Aline*40,Zone*2
      Integer Surfac,Test
      Logical AskQus
      DO 1 I = 1,Ncols
    1   Test(I) = Surfac(I)
      Nzone = 1
    5 Call Notice('The current surface is set to:',.true.,5)
      Call Vtab(7)
      Write(*,1000) (Test(I),I=1,Ncols)
 1000   Format(4(' 1',8X,' 6',8X),/,40(1X,I1))
   10 If (AskQus('Do you wish to alter the zone definition? (Y/N): N'
     &     //Char(8),.false.,10)) THEN
        Nzone = Nzone + 1
        If (Nzone.GT.5) THEN
          Call Warnin('There can only be 5 or fewer zones',.false.,12)
          Nzone = Nzone - 1
        ELSE
          Write(Zone,1005) Nzone
 1005       Format(I2)
   15     Call Notice('\\Enter 1st and last column for zone '//Zone//
     &          ': ',.false.,12)
          Read(*,*,Err=15) Ics1,Ics2
          If (Ics1.GT.Ics2.OR.Ics1.LT.1.OR.Ics2.GT.Ncols) Goto 15
          DO 20 I = Ics1,Ics2
   20       Test(I) = Nzone
          If (Ics2.LT.Ncols) THEN
            DO 25 I = Ics2+1,Ncols
   25         Test(I) = Nzone + 1
          ENDIF
        ENDIF
        Goto 5
      ENDIF
      DO 30 I = 1,Ncols-1
```

```
        If (Test(I).LE.0.OR.Test(I).GT.5.OR.Test(I).GT.Test(I+1))
     &     THEN
           Call Warnin('Zones are out-of-order or < 1 or > 5!',
     &          .false.,`6)
           Goto 5
         ENDIF
   30 Continue
      Do 35 I = 1,Ncols
   35   Surfac(I) = Test(I)
      Return
      END
      Subroutine Alter(Which,Data,Ncols,Nrows)
C-------------------------------------------------------------------------------
C-------------------------------------------------------------------------------
      Dimension Data(41,41)
      Character Which*6
      Integer Row,Col
      Logical AskQus
    1 Call Notice('Do you wish to specify a zone of different values',
     &            .true.,5)
      If (AskQus('for the '//Which//' matrix? (Y/N): N'//char(8),
     &           .false.,6)) THEN
   10   Call Vtab(8)
        Write(*,1000) ' 1st & last columns'
 1000     Format(' Please enter',A,': ',\)
        Read(*,*,Err=10) Ics1,Ics2
        If (Ics1.GT.Ics2.OR.Ics1.LT.1.OR.Ics2.GT.Ncols) Goto 10
   20   Call Vtab(10)
        Write(*,1000) ' 1st & last rows'
        Read(*,*,Err=20) Irs1,Irs2
        If (Irs1.GT.Irs2.OR.Irs1.LT.1.OR.Irs2.GT.Nrows) Goto 20
   30   Call Vtab(12)
        Write(*,1000) ' region''s value'
        Read(*,*,Err=30) Rval
        Call Vtab(15)
        Write(*,1010) Which,Ics1,Ics2,Irs1,Irs2,Rval
 1010     Format(' The ',A,' matrix from cols ',I2,'..',I2,
     &          ' & rows ',I2,'..',I2,' will be set to ',G12.5)
        If (AskQus('Is this correct? (Y/N): Y'//char(8),
     &        .false.,22)) THEN
          DO 40 Col = Ics1,Ics2
            DO 40 Row = Irs1,Irs2
   40         Data(Col,Row) = Rval
        ENDIF
        Goto 1
      ENDIF
      Return
      END
      Subroutine PrtDat(Title,Dx,Ncols,Nrows,Surfac,Seep,Drain,Head,
     &                  Solute)
C-------------------------------------------------------------------------------
C-------------------------------------------------------------------------------
      Dimension Head(41,41),Solute(41,41),Surfac(41),Seep(41),
     &          Drain(41)
      Integer Surfac,Seep,Drain
      Character Title*80,Finput*15,ALine*132
      Write(12,1010) Title
 1010   Format(A)
      Write(12,1020) Ncols,Nrows,Dx
 1020   Format(2I5,G12.5)
      Write(12,1030) (Surfac(I),I=1,Ncols)
 1030   Format(3X,10(5X,I2,5X))
      If (Ncols.GT.10) THEN
        Istop = 10
      ELSE
        Istop = Ncols
      ENDIF
      Do 130 I = 1,Nrows
        If (Ncols.GT.10) THEN
          Write(Aline,1035) Seep(I),(Head(J,I),J=1,10)
 1035       Format(1X,I1,1X,10G12.5)
          Call StrLen(Aline,Length)
          Write(12,1010) ALine(1:Length)
          Nstop = 10
  125     Nstop = Nstop + 10
          Nstart = Nstop - 9
          If (Nstop.GT.Ncols) Nstop = Ncols
          ALine = ' '
          Write(Aline,1040) (Head(J,I),J=Nstart,Nstop)
 1040       Format(3X,10G12.5)
          If (Nstop.LT.Ncols) THEN
            Call StrLen(Aline,Length)
            Wri  (12,1010) ALine(1:Length)
            Goto 125
          ELSE
            Call StrLen(Aline,Length)
            Write(Aline(Length+2:Length+2),1045) Drain(I)
 1045         Format(I1)
            Write(12,1010) Aline(1:Length+2)
          ENDIF
        ELSE
          Write(Aline,1035) Seep(I),(Head(J,I),J=1,Ncols)
          Call StrLen(Aline,Length)
          Write(Aline(Length+2:Length+2),1045) Drain(I)
```

```
        Write(12,1010) Aline(1:Length-2)
      ENDIF
130 Continue
    Do 135 I = 1,Nrows
135   Write(12,1040) (Solute(J,I),J=1,Ncols)
    Close(12)
    Return
    END
```

Glossary of Common Block Variables in TWODSWAP
Two-Dimensional Transient Water & Solute Flow Simulation

```
Bulkdn = Bulk Density (gm/cc)
Cact   = Total Solute which entered the system plus what was there originally
CapMin = Minimum value the water capacity will be allowed to assume
Cdissv = Solute concentration (meq/cc)--dimensioned as (2,0:MaxCol+1,0:MaxRow+1)
Cinit  = Total amount of solute at the start of the simulation
Cnow   = Current actual total solute concentration at a given point in time
Col    = Column index
ConSat = Saturated conductivity (cm/sec)
Cpred  = Total solute in the system plus that which left at a given point in time
Diagon = 'B' coefficient used in the Thomas algorithm -- dimensioned as (MaxCol)
Dispr0 = Intercept coefficient for dispersion equation
Dispr1 = Slope coefficient for dispersion equation
DranMn = Node at which the drain started
DranMx = Node at which the drain ends
Dt     = Current time interval (sec)
DtDown = Boolean which indicated that the Dt value is decreasing
Dtmax  = Maximum Dt value (sec)
Dtmin  = Minimum Dt Value (sec)
DtStar = Boolean which indicates that the Dt value is to be changed
DtStep = The increment that Dt increases
Dx     = Nodal spacing (cm)
ExpLin = Exponent used in non-linear retardation coefficients
Fbrkth = Filename of the Breakthrough file
FlowIn = Mass balance accumulators for water & solute input --dimensioned as (0:2)
FlowOu = Mass balance accumulators for water & solute output --dimensioned as (0:2)
Fluxtm = Times (secs) at which flux values change --dimensioned as (MaxSet)
Fmatrx = Filename of the file which will hold the output matrices
Fprint = Filename for the results reports
GravFc = Gravity factor (1=gravity on, 0 = off)
Head   = Pressure head (cm) --dimensioned as (3,0:MaxCol+1,0:MaxRow+1)
Ibrk   = Counter which tracks the number of points in the BTC curve
Icheck = The number of iterations between checks on external abort commands
IflxPt = Integer pointer which indicates which flux values are current
Irepet = Counter of the number of Dt Steps which have taken place
ItrLim = The maximum number of iterations attempted during any convergence attempt
ItrTrk = Array which tracks the relative frequency of various iterative values --dimensioned as (0:1,0:30)
LckDwn = Boolean which indicates that the Dt is fixed at DtMin
MaxCol = Maximum number of columns in the system
MaxRep = Maximum number of reports
MaxRow = Maximum number of rows
MaxSet = Maximum number of parameter sets
Ncols  = Number of columns in simulation
Nflux  = Number of flux changes occurring during the full simulation
Nrepr  = Number of reports occurring during the simulation
Nrows  = Number of rows
Nzone  = Surface zones (1..?) for a given column  --dimensioned as (Maxcol)
OpenDr = Boolean indicating whether the drain is open or not  --dimensioned as (MaxRows)
Qijmh  = Flux values (q*Dx/2) half a node above any node --dimensioned as (MaxCol,MaxRow)
Qijph  = Flux values (q*Dx/2) half a node below any node --dimensioned as (MaxCol,MaxRow)
Qimhj  = Flux values (q*Dx/2) half a node to the left of any node --dimensioned as (MaxCol,MaxRow)
Qiphj  = Flux values (q*Dx/2) half a node to the right of any node --dimensioned as (MaxCol,MaxRow)
Reflct = Boolean indicating that the surface is a reflection boundary
RepHed = Boolean indicating that the IFLXPT report is to include head values --dimensioned as (0:MaxRep)
RepSol = Boolean indicating that the IFLXPT report is to include absolute solute values --dimensioned as (0:MaxRep)
RepThe = Boolean indicating that the IFLXPT report is to include theta values --dimensioned as (0:MaxRep)
Result = 'D' coefficient of the Thomas Algorithm --dimensioned as (MaxCol)
Rlabel = Run label
Row    = Row index
RpConc = Boolean indicating that the IFLXPT report is to include  solute concentration values --dimensioned as (0:MaxRep)
Rtime  = Pointer indicating the next report
SbDiag = 'A' coefficient of the Thomas Algorithm --dimensioned as (MaxCol)
Sdissv = Temporary holder of solute values --dimensioned as (MaxCol,MaxRow)
SeepC  = IFLXPT concentration of solute at the seep --dimensioned as (MaxSet)
SeepMn = Node at which the seep started
SeepMx = Node at which the seep ended
SeepQ  = IFLXPT flux at the seep --dimensioned as (MaxSet)
Shead  = Temporary holder of head values --dimensioned as (MaxCol,MaxRow)
SlSrbC = Retardation coefficient
SolErr = Solute Error (%)
SolFix = Boolean indicating that the solute is frozen
SolMax = Maximum expected solute concentration anywhere in the system
Solute = Character strings of 'Water' and 'Solute' --dimensioned as (0:1)
SpDiag = 'C' coefficient of the Thomas Algorithm --dimensioned as (MaxCol)
Strict = Boolean indicating that the iterative tolerance criterion is applied without exception
SurfC  = IFLXPT concentration of solute at the surface in the indicated zone --dimensioned as (MaxSet,MaxSet)
SurfQ  = IFLXPT flux at the surface in the indicated zone --dimensioned as (MaxSet)(MaxSet,MaxSet)
Table  = Vector containing coefficients need to calculate Head-Theta-Conductivity relations --dimensioned as (5)
TDijmh = Theta*Dispersion value above any node --dimensioned as (MaxCol,MaxRow)
TDijph = Theta*Dispersion value below any node --dimensioned as (MaxCol,MaxRow)
TDimhj = Theta*Dispersion value to the left any node --dimensioned as (MaxCol,MaxRow)
TDiphj = Theta*Dispersion value to the right any node --dimensioned as (MaxCol,MaxRow)
ThetaS = Saturated Water content
Time   = Current time (secs)
Tlevel = Tolerance level (%) used to determine when two consecutive guess are close enough
TmLast = The time of the last report
TmRepr = The times at which reports are to be generated --dimensioned as (0:MaxRep)
TmSpan = The time between successive BTC curve points
Tmstop = The time to stop the simulation
Wact   = The initial water content plus all water that entered the system
WatErr = Error (%) for water (based on Theta)
WatFix = Boolean indicating that the Water is at a steady state
Winit  = The initial water content
Wnow   = The current water content
Wpred  = The current water content plus whatever left the system
```

```
C Common block for twodSolu
      Implicit Double Precision (A-H,O-Z)
      SAVE
      Parameter(MaxCol=51, MaxRow=51, MaxSet=5, MaxRep=10)
      Common/Comchr/Rlabel,Fprint,Fmatrx,Solute(0:1),Fbrkth
      Common/Contrl/CapMin,Dispr0,Dispr1,DtDown,Dtmin,Dtmax,DtStar,
     &              DtStep,Dt,Dx,ExpLin,Feedbk,GravFc,Ibrk,Icheck,
     &              Irepet,ItrLim,ItrTrk(0:1,0:30),LckDwn,Nrows,
     &              Ncols,OpenDr(MaxRow),Reflct,Rtime,SlSrbC,
     &              Strict,SolFix,SolMax,Time,Tlevel,Tmstop,WatFix
      Common/Zones/SeepMn,SeepMx,Nzone(Maxcol),DranMn,DranMx
      Common/Flux/Fluxtm(MaxSet),Nflux,IflxPt,TmSpan,
     &            SurfQ(MaxSet,MaxSet),SurfC(MaxSet,MaxSet),
     &            SeepQ(MaxSet),SeepC(MaxSet)
      Common/MassBl/FlowIn(0:2),FlowOu(0:2),WatErr,Winit,Wact,Wpred,
     &            Wnow,SolErr,Cinit,Cact,Cpred,Cnow
      Common/Matrix/Head(3,0:MaxCol+1,0:MaxRow+1),
     &              Cdissv(2,0:MaxCol+1,0:MaxRow+1),
     &              Shead(MaxCol,MaxRow),Sdissv(MaxCol,MaxRow),
     &              Qimhj(Maxcol,MaxRow),Qiphj(MaxCol,MaxRow),
     &              Qijmh(MaxCol,MaxRow),Qijph(MaxCol,MaxRow),
     &              TDimhj(Maxcol,MaxRow),TDiphj(MaxCol,MaxRow),
     &              TDijmh(MaxCol,MaxRow),TDijph(MaxCol,MaxRow)
      Common/RepDat/Nrepr,RepHed(0:MaxRep),RepSol(0:MaxRep),
     &              RepThe(0:MaxRep),TmRepr(0:MaxRep),TmLast,
     &              RpConc(MaxRep)
      Common/Soil/Bulkdn,Table(5),ThetaS,ConSat
      Common/TriDat/SbDiag(MaxCol),Diagon(MaxCol),SpDiag(MaxCol),
     &              Result(MaxCol)
      Character Rlabel*80,Fprint*15,Fmatrx*15,Solute*16,Fbrkth*15
      Integer Row,Col,Rtime,SeepMn,SeepMx,SurfMn,SurfMx,DranMn,DranMx
      Logical FeedBk,DtStar,RepHed,WatFix,RepThe,DtDown,RepSol,RpConc,
     &        SolFix,OpenDr,Reflct,Strict
```

```
      Program Flow2D
C-----------------------------------------------------------------------------------------------------------------------C
C                  Dr. Stephen A. Bloom - for Dr. Robert Mansell, Soil Science 08/03/88
C               This is a Transient 2-Dimensional Water-Flow Simulation Program with Retarded Solute
C-----------------------------------------------------------------------------------------------------------------------C
      INCLUDE  'Twodswap.cmb'
      Logical Redo,Abort,Stopit,FirOvr
      Data FirOvr/.true./
      Call Page('Two-Dimensional Transient Water & Solute Flow')        {initialize the program utility name}
      Call StartR                                                       {acquire the run parameters & initialize the system}
      Call SetQTD                                                       {initialize the flux and Theta-Dispersion values}
      Abort = Stopit()                                                  {initialize the external abort system}
      Call Report(.true.)                                               {record initial start conditions of col..
      Call Notice(Rlabel,.true.,3)
C-------REPEAT-------------------------------------
       Call ChkFlx                                                      {impose the initial flux/head conditions}
  10   Call DtChan                                                      {check whether Dt needs changing}
       Redo = .false.                                                 {don't have to redo the water when starting a Dt step}
       TIME = TIME + DT                                                 {increment the clock}
       IREPET = IREPET + 1                                              {increment the iteration counter}
       If (.not.Abort.AND.MOD(Irepet,Icheck).EQ.0) Abort = Stopit()     {check if external abort desired}
       Call ChkFlx                                                      {check for flux changes}
       IF (Time.GE.(Ibrk+1)*TmRepr(Nreps)/200.0.OR.Abort) Call BrkOut   {& save breakthru info periodically}
       IF (Time.GE.TMREPR(Rtime).OR.Abort) Call Report(.true.)          {check if a Report is needed}
  20   If (.Not.Abort.AND.Time.LT.TmStop) THEN                          {if the simulation is continuing...
         Call WatItr(Ipass)                                             {simulate the Water - head pressure}
         Call SetQTD                                                    {set the flux & theta-Dispersion values}
         Call Solitr(Redo,Jpass,CdMax)                                  {simulate the solute movement}
         If (Redo) Goto 20                                              {if solute failed, Dt changed, so redo Water}
         Call MBUpDt                                                    {update the mass balance trackers }
         If (Ipass.GT.0) ItrTrk(0,Ipass) = ItrTrk(0,Ipass) + 1          {note the number of iterations needed-water}
         If (Jpass.GT.0) ItrTrk(1,Jpass) = ItrTrk(1,Jpass) + 1          {note the number of iterations needed-solute}
         If (Feedbk.OR.MOD(Irepet,25).EQ.0.OR.Irepet.LE.15)Call Mbalck  {periodically Report the mass balance errors}
      If (CdMax.GT.SolMax.AND.FirOvr) THEN                              {if there has been an overrun of solute & its}
        Call Report(.false.)                                            {the first time, then generate a report}
        FirOvr = .false.
      ENDIF
        ENDIF
C-------UNTIL--------------------------------------
      IF (.Not.Abort.AND.Time.LT.TmStop) GOTO 10                        {until the terminal time is attained}
      Call BrkRep                                                       {complete the BTC file and Report the iterative}
      Call RepItr                                                       {traces as well as material leaving unexpectedly}
      Write(12,1050) FLowIn(2),FLowOu(2)
 1050   Format(/,' Negative accummulators: In=',G12.5,' Out=',G12.5)
      CLOSE(12)
      END
      SUBROUTine WatItr(Iterat)
C-----------------------------------------------------------------------------------------------------------------------
C This routine coordinates the iterative solution of the water equation
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE  'Twodswap.cmb'
      Logical DoMore,InSeep,Stable
      If (.Not.WatFix) THEN                                             {if the water system is still in 'flux'}
        Stable = .true.                                                 {initially set stable flag to true}
        DO 5 Row = 0,Nrows+1                                            {for all nodes - including imaginary ones}
          Do 5 Col = 0,Ncols+1
            Hchang = Head(3,Col,Row)-Head(1,Col,Row)                    {for each node, determine the N->N+1 difference}
            If (DABS(Hchang).GT.1.0D-06) Stable = .false.               {if it is 'significant', Stable is false}
            Head(1,Col,Row) = Head(3,Col,Row)                           {transfer the data from N+1 -> N and }
            Head(2,Col,Row) = Head(3,Col,Row) + Hchang/2.0D+00          {extrapolate the changes into the future as the}
    5       Head(3,Col,Row) = Head(3,Col,Row) + Hchang                  {first guess}
        If (Stable) THEN                                              {if the system was found to be stable (no head change)}
          Nstable = Nstable + 1                                         {increment the stability counter}
          If (Nstable.GE.5) THEN                                        {if it has happened 5 times in the row,...}
            WatFix = .true.                                           {conclude that the system has reached a steady state}
            Write(*,1005) Time,Irepet                                   {fix the water as it stands & notify the user}
            Write(12,1005) Time,Irepet
 1005        Format(' ---->STEADY WATER SYSTEM ACHEIVED at ',G12.5,I5)
          ENDIF
        ELSE
          Nstable = 0                                                   {otherwise set the stability counter to 0}
        ENDIF
      ENDIF
      If (WatFix) THEN                                                {if the system is fixed, iteration is 0, otherwise..}
        Iterat = 0
      ELSE
        Nopen = 0                                                     {the number of times the drain has attempted to open}
        Ntrys = 0                                                     {the number of trys at getting a convergent soution}
    9   Nopen = Nopen + 1
        Nreset = 0                                                  {get ready to track reset efforts & iterative failures}
   10   Nreset = Nreset + 1
   15   Ifail = 0
   20   Iterat = 0
   25     Iterat = Iterat + 1                                            {count the number of passes in trying to get a
convergence}}

          Ntrys = Ntrys + 1
          Do 30 Row = 1,Nrows                                           {for all real rows in the system, solve for heads }
   30       Call WatRow(Row)                                            {from N -> N+1}
          Do 35 Col = 1,Ncols                                           {then update the future with the results}
            Do 35 Row = 1,Nrows
   35         Head(3,Col,Row) = Shead(Col,Row)
          Call FxBndH                                                   {set the boundaries as required}
          DifMax = 0.0D+00
          Do 40 Col = 1,Ncols                                           {for all real columns in the system, solve for heads
from}
   40       Call WatCol(Col,DifMax)                                     {N -> N+1, update and set the boundaries as required}
          Do 45 Col = 1,Ncols
```

```
            Do 45 Row = 1,Nrows
   45         Head(3,Col,Row) = Shead(Col,Row)
            Call FxBndH
          If (Ntrys.GT.ItrLim*5) Return                          {if the total number of passes > 5*limit, give it up}
          If (DoMore(Iterat,DifMax)) Goto 25                       {if a convergent solution is not yet acheived, try
again}
          If (Iterat.GE.ItrLim) THEN                               {unless the iterative limit is exceeded.  In that
case...}
            Ifail = Ifail + 1                                    {increment the failure counter, change Dt to DtMin or}
            DtStar = .true.                                     {even lower if IFAIL > 1, set the failure counter, flag}
            Call DtChan                                            {the failure and try again at the new Dt}
            Dt = Dt/Ifail
            ItrTrk(0,0) = ItrTrk(0,0) + 1
            Write(*,1025) Dt,Time
 1025         Format(' Water Failed: Dt=',F8.3,' Time=',G12.5)
            If (Ifail.LT.2) Goto 20                                {but don't fail more then twice}
          ENDIF
        If (DranMx.GT.0) THEN                                      {if there is a drain....}
          DO 50 Row = DranMn,DranMx                                {for every row within the drain zone...}
            If (OpenDr(Row).AND.Head(3,Ncols,Row).LT.0.0D+00) THEN {if the drain is open but the head is negative}
              DtStar = .true.                                      {the head must have oscillated from + to -, then}
              Dt = Dt/2.0D+00                                      {halve the Dt and try again}
              If (Nreset.LE.5.OR.MOD(Nreset,5).EQ.0) Write(*,1030) {allow 5 attempts at damping the oscillation}
     &                        Time,Nreset,Dt
 1030          Format(' Negative overshoot at drain at ',F10.2,
     &           ' Nreset=',I2,' Dt=',G12.5)
              Goto 10
            ELSE If (.Not.OpenDr(Row)) THEN                        {if there is a drain and it is closed...}
              HedAvr = 0.5*(Head(1,Ncols,Row)+Head(3,Ncols,Row))   {check to see if it should open by examining the}
              If (HedAvr.GE.0.0D+00) THEN                         {the average head at the drain.  If is > 0.0 & < 1.0}
                If (HedAvr.LE.1.0D+00) THEN
                  OpenDr(Row) = .true.                             {open the drain (permanently), imposed a head of}
                  Head(3,Ncols,Row) = 0.0D+00
                  Head(3,Ncols+1,Row) = 0.0D+00                    {0 at the drain and notify the user of the action}
                  Write(*,*) ' Drain ',Row,' opened at iter. ',Irepet
                  Write(12,*) ' Drain ',Row,'  opened at iter. ',Irepet
                ELSE                                               {it overshot (got too positive in a single Dt), so
                  Write(*,*) ' Dt reduced in anticipation of drain ',  {reduce the Dt and try again}
     &                       Row,' opening'
                  Dt = Dt/2.0D+00
                  If (Nopen.LT.10) Goto 9                          {but only 9 times or fewer}
                  Opendr(Row) = .true.
                ENDIF
              ENDIF
            ENDIF
   50     Continue
        ENDIF
        Do 55 Col = 0,Ncols+1                                      {for all nodes, insure that head at N+1/2 is
          Do 55 Row = 0,Nrows+1                                    {the average of N and N+1}
   55       Head(2,Row,Col) =
     &            (Head(1,Row,Col) + Head(3,Row,Col))/2.0D+00
      ENDIF
      Return
      END
      SUBROUTINE WatRow(Irow)
C-------------------------------------------------------------------------------------------------------------------------C
C This routine solves the water equation row-wise (across columns for a given row)
C-------------------------------------------------------------------------------------------------------------------------C
      INCLUDE  'Twodswap.cmb'
      Logical InSeep,InDran
      DxSqDt = 2.0*Dx*Dx/Dt
      Do 40 Col = 1,Ncols                                          {for all internal nodes - determine the surrounding
        Cdiphj = AvrCon( Col , Irow ,Col+1, Irow ,2)               {conductivities as a function of head at n+1/2}
        Cdijph = AvrCon( Col , IRow , Col ,Irow+1,2)
        AlphaT = DXsqDt*Cslope(
     &             (Head(1,Col,IRow) + Head(3,Col,Irow))/2.0D+00 )   {slope term is average of current + guess}
        If (Inseep(Irow).AND.Col.EQ.1) THEN                        {Flux-Seep equation}
          Cdijmh = AvrCon( Col ,Irow-1, Col , Irow ,2)
          CdSum =  Cdiphj + Cdijmh + Cdijph
          SbDiag(1) = 0.0D+00
          Diagon(1) = AlphaT + CdSum
          SpDiag(1) = -Cdiphj
          Result(1) = (Cdijmh          )*Head(1,1,IRow-1) +
     &                (AlphaT - CdSum )*Head(1,1, Irow ) +
     &                (Cdijph         )*Head(1,1,IRow+1) +
     &                (Cdiphj         )*Head(1,2,Irow)   +
     &                (Cdijmh         )*Head(3,1,IRow-1) +
     &                (Cdijph         )*Head(3,1,IRow+1) +
     &          2.0*Dx*(Gravfc*(Cdijmh-Cdijph) + SeepQ(Iflxpt))
        ELSE If (.Not.Reflct.AND.Irow.EQ.1) THEN                   {Flux-Surface Equation}
          Cdimhj = AvrCon(Col-1, Irow , Col , Irow ,2)
          CdSum = Cdimhj + Cdiphj + Cdijph
          SbDiag(Col) = -Cdimhj
          Diagon(Col) =  AlphaT + CdSum
          SpDiag(Col) = -Cdiphj
          Result(Col) = (AlphaT - CdSum )*Head(1, Col ,1) +
     &                  (Cdijph         )*Head(1, Col ,2) +
     &                  (Cdimhj         )*Head(1,Col-1,1) +
     &                  (Cdiphj         )*Head(1,Col+1,1) +
     &                  (Cdijph         )*Head(3, Col ,2) +
     &       2.0*Dx*(-GravFc*Cdijph + SurfQ(Nzone(Col),Iflxpt))
        ELSE                                                       {normal internal equation}
          Cdimhj = AvrCon(Col-1, Irow , Col , Irow ,2)
          Cdijmh = AvrCon( Col ,Irow-1, Col , Irow ,2)
          CdSum = Cdimhj + Cdiphj + Cdijmh + Cdijph
          SbDiag(Col) = -Cdimhj
          SpDiag(Col) = -Cdiphj
          Diagon(Col) =  AlphaT + CdSum
          Result(Col) = 2.0*GravFc*DX*(Cdijmh - Cdijph) +
     &                  (Cdijmh         )*Head(1,Col,IRow-1) +
     &                  (AlphaT - CdSum )*Head(1,Col, Irow ) +
```

```
     &                    (Cdijpn          )*Head(1,Col,IRow+1) +
     &                    (Cdimhj          )*Head(1,Col-1,IRow) +
     &                    (Cdiphj          )*Head(1,Col+1,IRow) +
     &                    (Cdijmn          )*Head(3,Col,IRow-1) +
     &                    (Cdijph          )*Head(3,Col,IRow+1)
        ENDIF
  40 Continue
     SpDiag(1) = SbDiag(1) + SpDiag(1)
     SbDiag(1) = 0.0D+00
     If (InDran(Irow).AND.OpenDr(Irow)) THEN                    {special case for an open drain node}
       SbDiag(Ncols) = 0.0D+00
       Diagon(Ncols) = 1.0D+00
       Result(Ncols) = 0.0D+00
     ELSE
       SbDiag(Ncols) = SbDiag(Ncols) + SpDiag(Ncols)
     ENDIF
     SpDiag(Ncols) = 0.0D+00
     Call TriDim(Ncols)                                         {solve for next head array}
     Do 130 Col = 1,Ncols                                       {for all nodes in tridiagonal solution &}
 130   SHead(Col,IRow) = Result(Col)                            {transfer results to a holding array}
     Return
     END
     SUBROUTINE WatCol(Icol,DifMax)
C-------------------------------------------------------------------------------------------------------------------------C
C This routine performs the column calculations for the water equation
C-------------------------------------------------------------------------------------------------------------------------C
     INCLUDE  'Twodswap.cmb'
     Logical InSeep
     DxSqDt = 2.0*Dx*Dx/Dt
     Do 40 Row = 1,Nrows                                        {for all internal nodes }
       Cdijph = AvrCon( Icol , Row , Icol ,Row+1,2)
       Cdiphj = AvrCon( Icol , Row ,Icol+1, Row ,2)
       AlphaT = DXsqDt*Cslope(
     &              (Head(1,Icol,Row) + Head(3,Icol,Row))/2.0D+00)
       If (Icol.EQ.1.AND.InSeep(Row)) THEN                      {seep flux boundary}
         Cdijmh = AvrCon( Icol ,Row-1, Icol , Row ,2)
         CdSum =  Cdiphj + Cdijmh + Cdijph
         SbDiag(Row) = -Cdijmh
         Diagon(Row) =  AlphaT + CdSum
         SpDiag(Row) = -Cdijph
         Result(Row) = (AlphaT - CdSum )*Head(1,1, Row ) +
     &                 (Cdiphj         )*Head(1,2, Row ) +
     &                 (Cdijmn         )*Head(1,1,Row-1) +
     &                 (Cdijph         )*Head(1,1,Row+1) +
     &                 (Cdiphj         )*Head(3,2, Row ) +
     &        2.0*Dx*(GravFc*(Cdijmh - Cdijph) + SeepQ(Iflxpt))
       ELSE If (.Not.Reflct.AND.Row.EQ.1) THEN                  {surface flux boundary}
         Cdimhj = AvrCon(Icol-1, Row , Icol , Row ,2)
         CdSum = Cdimhj + Cdiphj + Cdijph
         SbDiag(1) = 0.0D+00
         Diagon(1) =  AlphaT + CdSum
         SpDiag(1) = -Cdijph
         Result(1) = (Cdimhj          )*Head(1,ICol-1,1) +
     &               (AlphaT - CdSum )*Head(1, ICol ,1) +
     &               (Cdiphj         )*Head(1,ICol+1,1) +
     &               (Cdijph         )*Head(1, ICol ,2) +
     &               (Cdimhj         )*Head(3,ICol-1,1) +
     &               (Cdiphj         )*Head(3,ICol+1,1) +
     &      2.0*Dx*(-GravFc*Cdijph + SurfQ(Nzone(Icol),Iflxpt))
       ELSE                                                     {normal internal equation}
         Cdimhj = AvrCon(Icol-1, Row , Icol , Row ,2)
         Cdijmh = AvrCon( Icol ,Row-1, Icol , Row ,2)
         CdSum = Cdimhj + Cdiphj + Cdijmh + Cdijph
         SbDiag(Row) = -Cdijmh
         Diagon(Row) =  AlphaT + CdSum
         SpDiag(Row) = -Cdijph
         Result(Row) = (Cdijmh          )*Head(1,ICol,Row-1) +
     &                 (AlphaT - CdSum )*Head(1,ICol, Row ) +
     &                 (Cdijph         )*Head(1,ICol,Row+1) +
     &                 (Cdimhj         )*Head(1,ICol-1,Row) +
     &                 (Cdiphj         )*Head(1,ICol+1,Row) +
     &                 (Cdimhj         )*Head(3,ICol-1,Row) +
     &                 (Cdiphj         )*Head(3,ICol+1,Row) +
     &                       2.0*GravFc*DX*(Cdijmh - Cdijph)
       ENDIF
  40 Continue
     If (Reflct) THEN                                           {if the surface is a reflection boundary}
       SpDiag(1) = SbDiag(1) + SpDiag(1)                        {then amnipulate the coefficients}
       Result(1) = Result(1) + Gravfc*Sbdiag(1)*Dx
       SbDiag(1) = 0.0D+00
     ENDIF
     Result(Nrows) = Result(Nrows) - Gravfc*Spdiag(Nrows)*Dx    {the bottom is always a reflection boundary}
     SbDiag(Nrows) = SbDiag(Nrows) + SpDiag(Nrows)
     SpDiag(Nrows) = 0.0D+00
     Call TriDim(Nrows)                                         {solve for next head array & scan for max error}
     Do 130 Row = 1,Nrows
       If (Head(3,Icol,Row).EQ.0.0D+00) THEN
         Error = 0.0D+00
       ELSE
         Error = 100.0D+00*DABS( (Result(Row)-Head(3,Icol,Row))/
     &                           Head(3,Icol,Row) )
       ENDIF
       If (Error.GT.DifMax) DifMax = Error
 130   SHead(Icol,Row) = Result(Row)                            {store results in a holder array}
     Return
     END
     SUBROUTINE SolItr(Redo,Iterat,CdMax)
C-------------------------------------------------------------------------------------------------------------------------
C This routine coordinates the iterative solution of the solute equation
C-------------------------------------------------------------------------------------------------------------------------
     INCLUDE  'Twodswap.cmb'
```

```
      Logical DoMore,InSeep,Redo
      If (SolFix) THEN                                          {if the solute is frozen, number of iterations=0}
        Iterat = 0
      ELSE                                                      {otherwise determine difference from n-1 to n,}
        DO 10 Row = 0,Nrows+1                                   {extrapolate from n->n+1/2 & n->n+1 for all nodes}
          Do 10 Col = 0,Ncols+1
            Cchang = Cdissv(2,Col,Row)-Cdissv(1,Col,Row)
            Cdissv(1,Col,Row) = Cdissv(2,Col,Row)
 10         Cdissv(2,Col,Row) = Cdissv(2,Col,Row) + Cchang
      If (.not.Redo) Ifail = 0                                  {if not trying to recvoer from an iterative failure}
 20   Iterat = 0
 25     Iterat = Iterat + 1                                     {calculate n+1 estimates row-wise, then column-wise}
        Do 30 Row = 1,Nrows                                     {updating the n+1 array from the holding array at}
 30       Call SolRow(Row)                                      {each half-step and reseting the boundaries}
        Do 35 Col = 1,Ncols
          DO 35 Row = 1,Nrows
 35         Cdissv(2,Col,Row) = Sdissv(Col,Row)
        Call FxBndS
        DifMax = 0.0D+00
        Do 40 Col = 1,Ncols
 40       Call SolCol(Col,DifMax)
        CdMax = 0.0D-00                                         {scan the internal nodes for the maximum solute value}
        Do 45 Col = 1,Ncols                                     {which will be checked against the largest expected}
          DO 45 Row = 1,Nrows                                   {value to test for overruns}
            If (Sdissv(Col,Row).GT.CdMax) CdMax = Sdissv(Col,Row)
 45         Cdissv(2,Col,Row) = Sdissv(Col,Row)
        Call FxBndS
        If (DoMore(Iterat,DifMax)) Goto 25                      {if the convergence has yet to occur, try again}
        If (Iterat.GE.ItrLim) THEN                              {but if the iterative limit was reached, reset}
          Ifail = Ifail + 1                                     {the Dt & redo the water solution from this time}
          DtStar = .true.                                       {step before coming back to this routine & retrying}
          Call DtChan                                           {the solute}
          ItrTrk(1,0) = ItrTrk(1,0) + 1
          Dt = Dt/Ifail
          Write(*,1020) Dt,Time
1020        Format(' Solute Failed : Dt=',F8.3,' at ',G12.5)
          If (Ifail.LT.2) THEN
            Redo = .true.
          ELSE
            Redo = .false.
          ENDIF
        ELSE
          Redo = .false.
        ENDIF
      ENDIF
      Return
      END
      SUBROUTINE SolRow(Irow)
C-----------------------------------------------------------------------------------------------------------------------C
C This routine solves row-wise (for a given row, across all columns)
C-----------------------------------------------------------------------------------------------------------------------C
      INCLUDE  'Twodswap.cmb'
      Logical InSeep
      DxSqDt = 2.0D+00*Dx*Dx/Dt
      Do 40 Col = 1,Ncols
        Alpha = DxSqDt*Theta(Head(2,Col,Irow))*Retard(Col,Irow)
        If (InSeep(Irow).AND.Col.EQ.1) THEN                     {flux seep equation}
          B1 = SeepQ(Iflxpt)*Dx/2.0D+00
          TDSum = TDiphj(1,Irow) + TDijmh(1,Irow) + TDijph(1,Irow)
     &      + 2.0D+00*B1 - Qiphj(1,Irow) + Qijmh(1,Irow) - Qijph(1,Irow)
          SbDiag(1) = 0.0D+00
          Diagon(1) = Alpha + TDSum
          SpDiag(1) = - TDiphj(1,Irow) + Qiphj(1,Irow)
          Result(1) =
     &       (TDijmh(1,Irow)+Qijmh(1,Irow))*CDissv(1,1,IRow-1) +
     &       (Alpha - TDSum                )*Cdissv(1,1, Irow ) +
     &       (TDijph(1,Irow)-Qijph(1,Irow))*Cdissv(1,1,IRow+1) +
     &       (TDiphj(1,Irow)-Qiphj(1,Irow))*Cdissv(1,2, IRow)+
     &       (TDijmh(1,Irow)+Qijmh(1,Irow))*CDissv(2,1,IRow-1) +
     &       (TDijph(1,Irow)-Qijph(1,Irow))*Cdissv(2,1,IRow+1) +
     &              4.0D+00*B1*SeepC(Iflxpt)
        ELSE If (.not.Reflct.AND.Irow.EQ.1) THEN                {flux surface equation}
          B3 = SurfQ(Nzone(Col),Iflxpt)*Dx/2.0D+00
          TDSum =  TDimhj(Col,1) + TDiphj(Col,1) + TDijph(Col,1) +
     &      Qimhj(Col,1) -  Qiphj(Col,1) + 2.0D+00*B3 - Qijph(Col,1)
          SbDiag(Col) = - TDimhj(Col,1) - Qimhj(Col,1)
          Diagon(Col) = Alpha + TDSum
          SpDiag(Col) = - TDiphj(Col,1) + Qiphj(Col,1)
          Result(Col) =
     &       (Alpha - TDSum              )*Cdissv(1, Col ,1) +
     &       (TDijph(Col,1)-Qijph(Col,1))*Cdissv(1, Col ,2) +
     &       (TDimhj(Col,1)+Qimhj(Col,1))*Cdissv(1,Col-1,1) +
     &       (TDiphj(Col,1)-Qiphj(Col,1))*Cdissv(1,Col+1,1) +
     &       (TDijph(Col,1)-Qijph(Col,1))*Cdissv(2, Col ,2) +
     &        4.0D+00*B3*SurfC(Nzone(Col),Iflxpt)
        ELSE                                                    {normal internal equation}
          TDSum = TDimhj(Col,Irow) + TDiphj(Col,Irow) +
     &            TDijmh(Col,Irow) + TDijph(Col,Irow) +
     &             Qimhj(Col,Irow) -  Qiphj(Col,IRow) +
     &             Qijmh(Col,Irow) -  Qijph(Col,Irow)
          SbDiag(Col) = - TDimhj(Col,Irow) - Qimhj(Col,Irow)
          Diagon(Col) = Alpha + TDSum
          SpDiag(Col) = - TDiphj(Col,Irow) + Qiphj(Col,Irow)
          Result(Col) =
     &       (TDijmh(Col,Irow)+Qijmh(Col,Irow))*CDissv(1,Col,IRow-1) +
     &       (Alpha - TDSum                )*Cdissv(1,Col, Irow ) +
     &       (TDijph(Col,Irow)-Qijph(Col,Irow))*Cdissv(1,Col,IRow+1) +
     &       (TDimhj(Col,Irow)+Qimhj(Col,Irow))*Cdissv(1,Col-1,IRow) +
     &       (TDiphj(Col,Irow)-Qiphj(Col,Irow))*Cdissv(1,Col+1,IRow) +
     &       (TDijmh(Col,Irow)+Qijmh(Col,Irow))*CDissv(2,Col,IRow-1) +
     &       (TDijph(Col,Irow)-Qijph(Col,Irow))*Cdissv(2,Col,IRow+1)
```

```
      ENDIF
   40 If (Result(Col).LT.-1.0D-08) Result(Col) = 1.0D-08
      SpDiag(1) = SbDiag(1) + SpDiag(1)                          {adjust for the Thomas algorithm}
      SbDiag(1) = 0.0D+00
      SbDiag(Ncols) = SbDiag(Ncols) + SpDiag(Ncols)
      SpDiag(Ncols) = 0.0D+00
      Call TriDim(Ncols)                                         {solve for next Cdissv array}
      Do 130 Col = 1,Ncols                                       {put results into a holding array}
        If (Result(Col).LT.0.0) Result(Col) = 0.0D+00
  130 Sdissv(Col,IRow) = Result(Col)
      Return
      END
      SUBROUTine SolCol(Icol,DifMax)
C-------------------------------------------------------------------------------------------------------------------------C
C This routine solves column-wise (for a given column down all the rows)
C-------------------------------------------------------------------------------------------------------------------------C
      INCLUDE  'Twodswap.cmb'
      Logical InSeep
      DxSqDt = 2.0D+00*Dx*Dx/Dt
      Do 40 Row = 1,Nrows
        Alpha = DxSqDt*Theta(Head(2,Icol,Row))*Retard(Icol,Row)
        If (Icol.EQ.1.AND.Inseep(Row)) THEN                      {flux seep equation}
          B1 = SeepQ(Iflxpt)*Dx/2.0D+00
          TDSum = TDiphj(1,Row) + TDijmh(1,Row) + TDijph(1,Row)
     &     + 2.0D+00*B1 - Qiphj(1,Row) + Qijmh(1,Row) -  Qijph(1,Row)
          SbDiag(Row) = -TDijmh(1,Row) - Qijmh(1,Row)
          Diagon(Row) = Alpha + TDSum
          SpDiag(Row) = -TDijph(1,Row) + Qijph(1,Row)
          Result(Row) =
     &         (Alpha - TDSum                    )*Cdissv(1,1, Row ) +
     &         (TDiphj(Icol,Row)-Qiphj(Icol,Row))*Cdissv(1,2, Row)+
     &         (TDijmh(Icol,Row)+Qijmh(Icol,Row))*Cdissv(1,1,Row-1) +
     &         (TDijph(Icol,Row)-Qijph(Icol,Row))*Cdissv(1,1,Row+1) +
     &         (TDiphj(Icol,Row)-Qiphj(Icol,Row))*Cdissv(2,2, Row)+
     &             4.0D+00*B1*SeepC(Iflxpt)
        ELSE If (.not.Reflct.AND.Row.EQ.1) THEN                  {flux surface equation}
          B3 = SurfQ(Nzone(ICol),Iflxpt)*Dx/2.0D+00
          TDSum = TDimhj(ICol,1) + TDiphj(ICol,1) + TDijph(ICol,1) +
     &     Qimhj(ICol,1) - Qiphj(ICol,1) + 2.0D+00*B3 - Qijph(ICol,1)
          SbDiag(Row) = 0.0D+00
          Diagon(Row) = Alpha + TDSum
          SpDiag(Row) = -TDijph(Icol,1) + Qijph(Icol,1)
          Result(Row) =
     &         (TDimhj(Icol,1)+Qimhj(Icol,1))*Cdissv(1,Icol-1,1) +
     &         (Alpha - TDSum                    )*Cdissv(1, ICol ,1) +
     &         (TDiphj(Icol,1)-Qiphj(Icol,1))*Cdissv(1,Icol+1,1) +
     &         (TDijph(Icol,1)-Qijph(Icol,1))*Cdissv(1, Icol ,2) +
     &         (TDimhj(Icol,1)+Qimhj(Icol,1))*Cdissv(2,Icol-1,1) +
     &         (TDiphj(Icol,1)-Qiphj(Icol,1))*Cdissv(2,Icol+1,1) +
     &             4.0D+00*B3*SurfC(Nzone(ICol),Iflxpt)
        ELSE                                                     {normal internal equation}
          TDSum = TDimhj(ICol,Row) + TDiphj(ICol,Row) +
     &            TDijmh(ICol,Row) + TDijph(ICol,Row) +
     &             Qimhj(ICol,Row) -  Qiphj(ICol,Row) +
     &             Qijmh(ICol,Row) -  Qijph(ICol,Row)
          SbDiag(Row) = -TDijmh(Icol,Row) - Qijmh(Icol,Row)
          Diagon(Row) = Alpha + TDSum
          SpDiag(Row) = -TDijph(Icol,Row) + Qijph(Icol,Row)
          Result(Row) =
     &         (TDimhj(Icol,Row)+Qimhj(Icol,Row))*Cdissv(1,Icol-1,Row) +
     &         (Alpha - TDSum                    )*Cdissv(1, ICol ,Row) +
     &         (TDiphj(Icol,Row)-Qiphj(Icol,Row))*Cdissv(1,Icol+1,Row) +
     &         (TDijmh(Icol,Row)+Qijmh(Icol,Row))*Cdissv(1,Icol,Row-1) +
     &         (TDijph(Icol,Row)-Qijph(Icol,Row))*Cdissv(1,Icol,Row+1) +
     &         (TDimhj(Icol,Row)+Qimhj(Icol,Row))*Cdissv(2,Icol-1,Row) +
     &         (TDiphj(Icol,Row)-Qiphj(Icol,Row))*Cdissv(2,Icol+1,Row)
        ENDIF
   40 If (Result(Row).LT.-1.0D-08) Result(Row) = 0.0D+00
      If (Reflct) THEN                                           {if the surface is a reflection boundary, so treat it}
        SpDiag(1) = SpDiag(1) + SbDiag(1)
        SbDiag(1) = 0.0D+00
      ENDIF
      SbDiag(Nrows) =  SbDiag(Nrows) + SpDiag(Nrows)
      SpDiag(Nrows) = 0.0D+00
      Call TriDim(Nrows)                                         {solve for next Cdissv array-scan for max error}
      Do 130 Row = 1,Nrows                                       {and place results into the holder array}
        If (Result(Row).LT.0.0) Result(Row) = 0.0D+00
        If (DABS(Result(Row)-Cdissv(2,Icol,Row)).LE.1.0D-06.OR.
     &   Cdissv(2,Icol,Row).EQ.0.0) THEN
          Error = 0.0D+00
        ELSE
          Error = 100.0D+00*DABS(
     &       (Result(Row)-Cdissv(2,Icol,Row))/Cdissv(2,Icol,Row) )
        ENDIF
        If (Error.GT.DifMax) DifMax = Error
  130   Sdissv(Icol,Row) = Result(Row)
      Return
      END
      SUBRoutIne MBUpDt
C-------------------------------------------------------------------------------------------------------------------------C
C This routine updates the mass balance accumulators
C-------------------------------------------------------------------------------------------------------------------------C
      INCLUDE  'Twodswap.cmb'
      Logical Abort
      Abort = .false.
      If (SeepMx.GT.0) THEN                                      {if there is a seep - for all rows within the seep}
        Do 10 Row = SeepMn,SeepMx
          WatIn = Dx*Dt*SeepQ(Iflxpt)                            {determine amount of water entering}
          WatMax = Dx*Dx*(ThetaS-Theta(Head(2,1,Row)))           {check for exceeding the available}
          If (WatMax.LT.WatIn) THEN                              {receptive capacity of the soil at the seep}
            Write(12,2000) 'Seep',WatIn,WatMax                   {if it does, sound a warning}
```

```
3002        Format(' Flux boundary error at ',A,' Water entering=',
     &       G12.5,' Capacity left=',G12.5,1X,A,I3)
            Abort = .true.
          ENDIF
          FlowIn(0) = FlowIn(0) + WatIn                              {accumulate the amount of water entering}
   10     FlowIn(1) = FlowIn(1) + WatIn*SeepC(Iflxpt)                {and the solute brought along with it}
        ENDIF
        IF (DranMx.GT.0) THEN                                        {if there is a drain..for all rows within it..}
          Do 20 Row = DranMn,DranMx
            If (OpenDr(Row)) THEN
              WatOut=-Dt*ConSat*(Head(2,Ncols+1,Row)-Head(2,Ncols,Row))   {calculate outflowing flux and check for reversal}
              If (WatOut.LT.0.0D+00) THEN
                Write(*,1010) Time,WatOut,
     &                        Head(2,Ncols,Row),Head(2,Ncols+1,Row)
1010              Format(' Water entering at drain at ',F10.2,' = ',
     &                        G12.5,' H=',2G12.5)
                FlowOu(2) = FlowOu(2) + WatOut
                WatOut = 0.0D+00
              ENDIF
              FlowOu(0) = FlowOu(0) + WatOut                         {accumulate the water leaving & the average }
              FlowOu(1) = FlowOu(1) + WatOut *                       {solute carried out with it}
     &          (Cdissv(2, Ncols ,Row) + Cdissv(1, Ncols ,Row))/2.0D+00
            ENDIF
   20     Continue
        ENDIF
        Do 25 Col = 1,Ncols                                          {across the surface, calculate water entering}
          WatIn = Dx*Dt*SurfQ(Nzone(Col),Iflxpt)
          WatMax = Dx*Dx*(ThetaS-Theta(Head(2,Col,1)))               {and check for exceeding available capacity}
          If (WatMax.LT.WatIn) THEN
            Write(12,2000) 'Surface',WatIn,WatMax,' Column=',col
            Abort = .true.
          ENDIF
          FlowIn(0) = FlowIn(0) + WatIn                              {accumulate entering water and solute}
   25     FlowIn(1) = FlowIn(1) + WatIn*SurfC(Nzone(Col),Iflxpt)
C        If (Abort) THEN                                            {this code will aid in auto-termination}
C          TmStop = Time - Dt
C          Call Report(.false.)
C        ENDIF
        Abort = .false.
        Return
        END
        SUBROUTINE SetQTD
C-------------------------------------------------------------------------------------------------------------------------
C This routine sets the flux arrays & theta-Dispersion arrays around all internal nodes
C-------------------------------------------------------------------------------------------------------------------------
        INCLUDE  'Twodswap.cmb'
        Logical InSeep
        Qalter = Dx/2.0D+00
        Do 20 Ir = 1,Nrows
         DO 20 Ic = 1,Ncols
           IF (Ic.EQ.1.AND.InSeep(Ir)) THEN                          {if at the seep, flux in is the seep flux}
             Qimhj(Ic,Ir) = SeepQ(Iflxpt)                            {and dispersion is 0}
             TDimhj(Ic,Ir) = 0.0D+00
           ELSE
             TDimhj(Ic,Ir) = ThetaD(Qimhj(Ic,Ir),Ic-1,Ir,Ic,Ir)      {otherwise calculate the flux from the left}
             Qimhj(Ic,Ir) = Qimhj(Ic,Ir)*Qalter
           ENDIF
           TDiphj(Ic,Ir) = ThetaD(Qiphj(Ic,Ir),Ic,Ir,Ic+1,Ir)        {the flux and 0-D (Theta-Dispersion) to the right}
           Qiphj(Ic,Ir) = Qiphj(Ic,Ir)*Qalter
           If (.Not.Reflct.AND.Ir.EQ.1) THEN                         {if there is a flux surface boundary}
             Qijmh(Ic,Ir) = SurfQ(Nzone(Ic),Iflxpt)*Qalter           {then the flux above is the surface flux & 0-D=0}
             TDijmh(Ic,Ir) = 0.0D+00
           ELSE
             TDijmh(Ic,Ir) = ThetaD(Qijmh(Ic,Ir),Ic,Ir-1,Ic,Ir)      {else calculate the flux & 0-D from above}
             Qijmh(Ic,Ir) = Qijmh(Ic,Ir)*Qalter
           ENDIF
           TDijph(Ic,Ir) = ThetaD(Qijph(Ic,Ir),Ic,Ir,Ic,Ir+1)        {calculate the flux & 0-D from below}
   20      Qijph(Ic,Ir) = Qijph(Ic,Ir)*Qalter
        Return
        END
        Double Precision Function ThetaD(Qflux,Icol,Irow,Jcol,Jrow)
C-------------------------------------------------------------------------------------------------------------------------
C This routine calculates the Theta-Disperison product between any pair of nodes
C-------------------------------------------------------------------------------------------------------------------------
        INCLUDE  'Twodswap.cmb'
        HeadAv = (Head(2,Icol,Irow) + Head(2,Jcol,Jrow))/2.0         {get the average head in space & time}
        Qflux = FLuxV1(Icol,Irow,Jcol,Jrow)                          {the flux between the nodes & the Theta}
        ThetaX = Theta(HeadAv)*Dispr0
        FluxX  = Dispr1*DABS(Qflux)                                  {0D = 0( Do + D1*Q/0) where 0=theta}
        ThetaD = ThetaX + FluxX
        If (DABS(Dispr0).LE.1.0D-06.AND.Dispr1.LE.1.0D-06)           {unless all 0D are to be zero}
     &        ThetaD = 0.0D+00
        Return
        END
        Logical Function OutSys(It,Ic,Ir)
C-------------------------------------------------------------------------------------------------------------------------
C This function determines if a given point in time and space is outside of the system & therefore in error
C-------------------------------------------------------------------------------------------------------------------------
        INCLUDE  'Twodswap.cmb'
        Logical InSeep,InDran
        If (It.LT.1.OR.It.GT.3.OR.IC.LT.0.OR.IC.GT.Ncols+1.OR.       {no question of illegality here}
     &    Ir.LT.0.OR.Ir.GT.Nrows+1) THEN
          OutSys = .true.
        ELSE
          If (Ic.EQ.0) THEN                                          {0 columns are okay if NOT in the seep}
            If (InSeep(Ir)) THEN
              OutSys = .true.
            ELSE
              OutSys = .false.
            ENDIF
          ELSE If (.not.Reflct.AND.Ir.EQ.1) THEN                     {0 rows OK if surface is a reflection}
```

```
        OutSys = .true.
      ELSE
        OutSys = .false.
      ENDIF
    ENDIF
    Return
    END
    SUBROUTINE GtHTab(Ftable)
C-------------------------------------------------------------------------------------------------------------------------
C This routine reads in the basic layout & head boundary conditions
C-------------------------------------------------------------------------------------------------------------------------
    INCLUDE  'Twodswap.cmb'
    Logical Error
    Character*(*) Ftable,Aline*1
    Integer SeepNd,DranNd
    Call OpnFil(Ftable,15,'OLD',Error)
    If (Error) Call ENdPgm('Unable to open MHD file')
    Read(15,1000) Aline                                          {file comment - never used in the program}
1000  Format(A)
    Read(15,*) Ncols,Nrows,Dx
    If (Nrows.GT.MaxRow) Call EndPgm('Too many rows required')
    If (Ncols.GT.MaxCol) Call EndPgm('Too many columns required')
    Read(15,*,Err=20,END=20) (Nzone(Col),Col=1,Ncols)            {surface zone description}
    Do 5 I = 1,Ncols-1
      If (Nzone(I).GT.Nzone(I+1)) Call ENdPgm('Zones out of order')
5     If (Nzone(I).LT.1) Call EndPgm('Zones must start with 1')
    If (Nzone(Ncols).Gt.MaxSet) Call ENdPgm('Too many zones')
    SeepMn = 0
    SeepMx = 0
    DranMn = 0
    DranMx = 0
    Do 10 Row = 1,Nrows
      Read(15,*,Err=20,END=20) SeepNd,(Head(3,Col,Row),Col=1,Ncols),  {read seep/head...head/drain info}
     &                         DranNd
      If (SeepNd.NE.1) THEN                                     {set the seep if indicated (1=no,0=yes)}
        If (SeepMn.EQ.0) SeepMn = Row
        SeepMx = Row
      ENDIF
      If (DranNd.NE.1) THEN
        If (DranMn.EQ.0) DranMn = Row
        DranMx = Row
      ENDIF
10  Continue
    If (SeepMn.LE.1.AND.SeepMx.GT.0) Call EndPgm('Seep incorrect')
    Do 15 Row = 1,Nrows                                         {read the solute concentrations}
      Read(15,*,Err=20,End=20) (Cdissv(2,Col,Row),Col=1,Ncols)
      Do 15 Col = 1,Ncols
15      If (Cdissv(2,Col,Row).GT.SolMax) SolMax = Cdissv(2,Col,Row)
    Close(15)
    Goto 25
20  Call EndPgm('problem with matrix table')
25  Return
    END
    SUBROUTINE ColSet
C-------------------------------------------------------------------------------------------------------------------------
C This routine sets the column according to the input controls
C-------------------------------------------------------------------------------------------------------------------------
    INCLUDE  'Twodswap.cmb'
    Call FxBndH                                                 {impose the boundary conditions & then set}
    Call FxBndS                                                 {all head and solute arrays identically}
    Do 20 Row = 0,Nrows+1
      Do 20 Col = 0,Ncols+1
        Cdissv(1,Col,Row) = Cdissv(2,Col,Row)
        Do 20 Itype = 1,2
20        Head(Itype,Col,Row) = Head(3,Col,Row)
    Return
    END
    SUBROUTINE ChkFlx
C-------------------------------------------------------------------------------------------------------------------------
C This routine handles setting the flux & velocity if they should change during the run
C-------------------------------------------------------------------------------------------------------------------------
    INCLUDE  'Twodswap.cmb'
    IF ((Iflxpt.LE.0).OR.                                       {if this is the first time this routine is called or
   &    (Time.GE.Fluxtm(Iflxpt+1).AND.(Iflxpt+1).LE.NFLUX)) THEN  {it is time to reset the flux values...}
      Iflxpt = Iflxpt + 1                                       {increment the flux ptr and reset the flux & veloc.}
      WatFix = .false.
      Write(12,1000) 'Imposed Flux',Iflxpt,Time
1000    Format(/,'->',A,' #',I2,' at ',F10.1,'sec.')
      DtStar = .true.
      If (SeepMx.GT.0) Write(12,1010) '  Seep','Imposed Flux',
   &                                  SeepQ(Iflxpt),SeepC(Iflxpt)
      Do 10 I = 1,Nzone(Ncols)
10      Write(12,1010)'Surface','Imposed Flux',
   &                            SurfQ(I,Iflxpt),SurfC(I,Iflxpt)
1010    Format(1X,A,': ',A,'=',G12.5,' Solute conc.=',G12.5)
      If (SolMax.LE.0.0D+00) THEN
        SolFix = .true.
        Write(*,*) ' Solute frozen'
      ELSE
        SolFix = .false.
        Write(*,*) ' Solute activated'
      ENDIF
      LckDwn = 0
    ENDIF
    LckDwn = LckDwn + 1
    Return
    END
    SUBROUTINE DtCHan
C-------------------------------------------------------------------------------------------------------------------------
C This routine resets the Dt
C-------------------------------------------------------------------------------------------------------------------------
```

```
      INCLUDE  'Twodswap.cmb'
      Logical DtAltr
      If (DtStep.GE.1.0) THEN
        Dt = DtMax
        Return
      ENDIF
      If (LckDwn.LE.10) THEN                                    {keep Dt = Dtmin fo the 1st 10 iterations after}
        Dt = Dtmin                                              {a flux change or when starting}
      ELSE
        DtNow = Dt                                              {otherwise increment it until Dt= DtMax}
        If (Iflxpt.GT.0.AND.Iflxpt.LT.Nflux) THEN               {but try to step down to Dtmin if a change is coming}
          Nsteps = NINT(-1.0 + (Fluxtm(Iflxpt+1)-(Time+Dt))/(Dt/2.0))
          If (Nsteps.LT.0) Nsteps = 0
          If (Nsteps.LT.20.AND.Nsteps.GT.0) THEN
            Dt = Dt*Nsteps*DtStep + DtMin
            DtDown = .true.
          ENDIF
        ENDIF
        If (DtStar.AND.LckDwn.GT.10) THEN                       {either true initially or when there is a flux change}
          DtStar = .false.
          DtAltr = .true.                                       {prevent entering here until next flux change, ready
          DtDown = .false.
          Dt = Dtmin - DtStep*DtMax                             {Error and set Dt to the minimum again}
        ENDIF
        IF (DTALTR.OR.DtDown) THEN
          If (.not.DtDown) DT = DT + DtStep*DtMax               {increment Dt}
          If (Dt.LE.DtMin+DtStep*DtMax) Dt = DtMin+DtStep*DtMax
          IF (DT.GE.DTMax) THEN                                 {up to the maximum limit set by the parameter file}
            DT = DTMax
            DTALTR = .FALSE.
          ENDIF
          Write(*,1010) DtNow,DT,IREPET
 1010       FORMAT(12X,'Dt changed from ',F10.3,' to ',F10.3,
     &                    ' at Iter.',I4)
        ENDIF
      ENDIF
      If (Time+Dt.GT.TmStop) Dt = TmStop - Time                 {make the last step come out even with termination}
      Return
      END
      Logical Function DoMore(iterat,DifMax)                    {SUBRoutine flag}
C-----------------------------------------------------------------------------------------------------------------------C
C This routine tests for iteration termination conditions
C-----------------------------------------------------------------------------------------------------------------------C
      INCLUDE  'Twodswap.cmb'
      Logical Recycl
      If ((iterat.EQ.1) .OR.
     &    (iterat.LT.ItrLim.AND.DifMax.GT.TLevel)) THEN
        Recycl = .true.
      ELSE
        Recycl = .false.
      ENDIF
      If (.Not.Strict.AND.iterat.GT.5.AND.Recycl) THEN
        If (DABS(DifMax-OldErr)/OldErr.LE.Tlevel.AND.
     &                      DifMax.LE.OldErr) Recycl = .false.
      ENDIF
      OldErr = DifMax
      DoMore = Recycl
      Return
      END
      Double Precision Function AvHead(Itime,Icol,Irow)
C-----------------------------------------------------------------------------------------------------------------------
C This routine returns average head values
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE  'Twodswap.cmb'
      Logical OutSys
      If (OutSys(itime,Icol,Irow)) THEN
        Write(*,1000) Itime,Icol,Irow
 1000     Format(' AvHead Reference Error: It,Ic,ir=',3I3)
        AvHead = 1.0D+30
      ELSE
        If (Itime.NE.2) THEN
          AvHead = Head(Itime,Icol,Irow)
        ELSE
          AvHead = (Head(1,Icol,Irow) + Head(3,Icol,Irow) )/2.0D+00
        ENDIF
      ENDIF
      Return
      END
      Double Precision Function AvrCon(Icol,Irow,Jcol,Jrow,Itime)
C-----------------------------------------------------------------------------------------------------------------------
C This routine returns average ConDuctivity between two nodes at two times
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE  'Twodswap.cmb'
      Logical InSeep,OutSys
      If (OutSys(Itime,Icol,Irow).OR.OutSys(Itime,Jcol,Jrow)) THEN
        AvrCon = 1.0D+30
        Write(*,1000) Icol,Irow,Jcol,Jrow,Itime
 1000     Format(' AvrCon Reference Error: Ic..jr,it=',5I3)
      ELSE
        AvrHed = 1.0D+30
        If (Irow.EQ.0) THEN
          AvrHed = AvHead(Itime,Icol,1)
        ELSE If (InSeep(Ir ).AND.Icol.EQ.0) THEN
          AvrHed = AvHead(Itime,1,Irow)
        ELSE
          AvrHed = (AvHead(Itime,Icol,Irow)+AvHead(Itime,Jcol,Jrow))/
     &                    2.0D+00
        ENDIF
        AvrCon = ConDuc(AvrHed)
      ENDIF
      Return
```

```
      END
      Double Precision Function FLuxV1(Icol,Irow,Jcol,Jrow)
C-----------------------------------------------------------------------------------------------------------------------
C This routine returns the flux between any two  nodes
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE  'Twodswap.cmb'
      Logical InSeep,OutSys
      If (IABS(Icol-Jcol)+IABS(Irow-Jrow).NE.1)
     &   Call EndPgm(' FluxV1: Arguments span > 1 Dx')
      If (OutSys(2,Icol,Irow).OR.OutSys(2,Jcol,Jrow)) THEN
        FluxVx = 1.0D+30
        If (InSeep(Irow).AND.Irow.EQ.Jrow.AND.Icol.EQ.0) THEN
          FluxVx = SeepQ(Iflxpt)
        ELSE If (Irow.EQ.0.AND.Icol.EQ.Jcol) THEN
          FluxVx = SurfQ(Nzone(Icol),Iflxpt)
        ENDIF
      ELSE
        If (Icol.EQ.Jcol) THEN
          Gfactr = 1.0D+00*GravFc
        ELSE
          Gfactr = 0.0D+00
        ENDIF
        HeadAv = (Head(2,Icol,Irow) + Head(2,Jcol,Jrow))/2.0D+00
        FLuxVx = -ConDuc(HeadAv)*
     &   ((Head(2,Jcol,Jrow)-Head(2,Icol,Irow))/Dx-Gfactr)
        If (DABS(FluxVX).LE.1.0E-10) FluxVx = 0.0D+00
      ENDIF
      FluxV1 = FluxVx
      Return
      END
      SUBROUTine Report(Normal)
C-----------------------------------------------------------------------------------------------------------------------C
C This routine  updates the plot/output file to be used by data presentation Programs. It Writes the depth, time,
C  head, water content, solute concentration, water flux & solute flux in a F7.1,F9.1,5G12.4 format.
C-----------------------------------------------------------------------------------------------------------------------C
      INCLUDE  'Twodswap.cmb'
      Character ALine*132,DranCh(MaxRow)*1,SeepCh(MaxRow)*1
      Logical InSeep,InDran,Normal
      Data Aline/' '/,Drancn/MaxRow*'+'/,SeepCh/MaxRow*'+'/
      Call Mbalck
      Hour = Time/3600
      Write(12,1000)Rlabel,Time,Hour,DX,Dt,Nrows,Ncols,GravFc,Irepet    {summation mass balance Report
 1000   Format(1X,57('-'),'<   Report    >',57('-'),//,24X,A,//,
     &    11X,'Time=',F10.2,' sec.=',F6.1,' hrs.; Dx=',F5.2,
     &    '; Dt=',F7.2,'; Row=',I2,' Col=',I2,' Gravity=',F2.0,
     &    ' Total ite-ations=',I9)
      Write(12,1005) FlowIn(0),FlowOu(0),Wact,Wpred,Waterr
 1005  Format(' ---[WATER]--- Total Flow In = ',G12.5,
     &   ' Flow out = ',G12.5,/,14X,' Sum of input & initial water = ',
     &   G12.5,' Sum of stored & outFlow = ',G12.5,
     &   ' Error%=-------(',G12.5,')-----')
      Write(12,1015) FlowOu(1),FlowIn(1),Cact,Cpred,SolErr
 1015   FORMAT(' ---[SOLUTE]-- Total Flow out = ',G12.5,/,
     &    ' in=',G12.5,' Sum of input & initial=',G12.5,
     &    ' of stored & outflow=',G12.5,' Error%=---(',G12.5,')---')
      Write(12,1016)
 1016    Format(' +',129(1H-),'+',/)
      Do 5 Row = 1,Nrows
        SeepCh(Row) = '|'
        DranCh(Row) = '|'
        If (InSeep(Row)) SeepCh(Row) = '>'
    5   If (InDran(Row)) DranCh(Row) = '>'
      Call StrLen(Rlabel,Length)
      If (Length.LE.0) Length = 1
      If (Ncols.GT.10) THEN
        Nstop = 10
      ELSE
        Nstop = Ncols
      ENDIF
      If (.not.Normal.OR.RepHed(Rtime)) THEN
        Write(14,1020) '{Head}'//Rlabel(1:Length),Ncols,Nrows,Dx,Time,
     &       WatErr
 1020     Format(A,/,2I4,3G12.5,' Rows,Columns,Dx,Current time,Error')
        Do 15 Row = 1,Nrows
          Write(14,1025) Row,SeepCh(Row),DranCh(Row),
     &                        (Head(3,Col,Row),Col=1,NStop)
   15     If (Nstop.LT.Ncols) Write(14,1030)
     &          (Head(3,Col,Row),Col=Nstop+1,Ncols)
 1025       Format(I3,1X,A,1X,A,1X,10G12.5)
 1030       Format(8X,10G12.5)
      ENDIF
      If (RepThe(Rtime)) THEN
        Write(14,1020) '{Theta}'//Rlabel(1:Length),Ncols,Nrows,Dx,Time,
     &          WatErr
        Do 25 Row = 1,Nrows
          Write(14,1025) Row,SeepCh(Row),DranCh(Row),
     &                       (Theta(Head(3,Col,Row)),Col=1,Nstop)
   25     If (Nstop.LT.Ncols) Write(14,1030)
     &              (Theta(Head(3,Col,Row)),Col=Nstop+1,Ncols)
      ENDIF
      If (.not.Normal.OR.RepSol(Rtime)) THEN
        Write(14,1020) '{Salt Conc}'//Rlabel(1:Length),Ncols,Nrows,Dx,
     &       Time,SolErr
        Do 35 Row = 1,Nrows
          Write(14,1025) Row,SeepCh(Row),DranCh(Row),
     &                      (Cdissv(2,Col,Row),Col=1,Nstop)
   35     If (Nstop.LT.Ncols) Write(14,1030)
     &                   (Cdissv(2,Col,Row),Col=Nstop+1,Ncols)
      ENDIF
      If (RpConc(Rtime)) THEN
        Write(14,1020) '{Salt Absl}'//Rlabel(1:Length),Ncols,Nrows,Dx,
```

```
     &       Time,SolErr
        Do 40 Row = 1,Nrows
          Write(14,1025) Row,SeepCh(Row),DranCh(Row),
     &         (Theta(Head(3,Col,Row))*Cdissv(2,Col,Row),Col=1,Nstop)
   40     If (Nstop.LT.Ncols) Write(14,1030)
     &    (Theta(Head(3,Col,Row))*Cdissv(2,Col,Row),Col=Nstop+1,Ncols)
        ENDIF
        If (DABS(SlSrbC).GT.1.0E-06.AND.RepSol(Rtime)) THEN
          Write(14,1020) '(Sorb Conc)'//Rlabel(1:Length),Ncols,Nrows,Dx,
     &       Time,SolErr
          Do 45 Row = 1,Nrows
            Write(14,1025) Row,SeepCh(Row),DranCh(Row),
     &           (SlSrbC*Cdissv(2,Col,Row)**ExpLin,Col=1,Nstop)
   45       If (Nstop.LT.Ncols) Write(14,1030)
     &           (SlSrbC*Cdissv(2,Col,Row)**ExpLin,Col=Nstop+1,Ncols)
        ENDIF
        If (Normal) Rtime = Rtime + 1
        Return
        END
        SUBROUTINE TriDim(Nodes)
C-------------------------------------------------------------------------------------------------------------------------------
C This routine solves a tri-diagonal matrix
C-------------------------------------------------------------------------------------------------------------------------------
        INCLUDE  'Twodswap.cmb'
        Dimension A(MaxCol),BETA(MaxCol),Y(MaxCol)
        A(1) = DIAGON(1)
        BETA(1)  = SPDIAG(1)/A(1)
        Y(1)     = RESULT(1)/A(1)
        DO 201 I = 2,Nodes
          Xt(I) = Result(I)
          A(I)  = DIAGON(I) - SBDIAG(I) * BETA(I-1)
          BETA(I)   = SPDIAG(I)/A(I)
  201     Y(I)      =(RESULT(I)-SBDIAG(I)*Y(I-1))/A(I)
        Result(Nodes) = Y(Nodes)
        DO 203 I = 1,Nodes-1
          J = Nodes -I
          Result(J) = Y(J) - BETA(J) * Result( J+1)
  203     If (Result(J).GT.-1.0E-10.AND.Result(J).LT.1.0E-10)
     &      Result(J) = 0.0D+00
        RETURN
        END
        Subroutine TotSys()
C-------------------------------------------------------------------------------------------------------------------------------C
C  sum theta & Solute throughout the system
C-------------------------------------------------------------------------------------------------------------------------------C
        INCLUDE  'Twodswap.cmb'
        Wnow = 0.0D+00
        Cnow = 0.0D+00
        DxSq = Dx*Dx
        DO 10 Row = 1,Nrows
          Do 10 Col = 1,Ncols
            ThetaX = Theta(Head(3,Col,Row))
            Wnow = Wnow + ThetaX*DxSq
   10       Cnow = Cnow + DxSq*Cdissv(2,Col,Row)*(ThetaX+BulkDn*SlSrbC)
        RETURN
        END
        SUBROUTINE Mbalck
C-------------------------------------------------------------------------------------------------------------------------------
C This routine performs a Mass Balance check for the Dt change monitoring procedure
C-------------------------------------------------------------------------------------------------------------------------------
        INCLUDE  'Twodswap.cmb'
        Call TotSys()
        If (Winit.LE.0.0) Winit = Wnow
        Wact = Winit + FlowIn(0)                                  {amount added to & in column to begin with
        Wpred = Wnow + FlowOu(0)                                  {amount left or still in column now
        WatErr = 100.0D+00 *(Wpred-Wact)/Wact                     {percent Error between the two
        If (Cinit.LE.0.0.AND.Irepet.LE.0) Cinit = Cnow
        Cact = Cinit + FlowIn(1)
        Cpred = Cnow + FLowOu(1)
        If (Cact.LE.0.0) THEN
          SolErr = 0.0D+00
        ELSE
          SolErr = 100.0D+00 *(Cpred-Cact)/Cact                    {percent Error between the two
        ENDIF
        Write(*,1000) Time,Waterr,SolErr
 1000     Format(1X,'At ',F10.1,' sec, Error %: Water=',G12.5,
     &         ' Solute=',G12.5)
        RETURN
        END
        SUBROUTINE RepItr
C-------------------------------------------------------------------------------------------------------------------------------
C This routine Reports the iterative values
C-------------------------------------------------------------------------------------------------------------------------------
        INCLUDE  'Twodswap.cmb'
        Character Aline*132
        Do 20 Itype = 0,1
          Write(12,1000) 'Iteration Trace:',Solute(Itype)
 1000       Format(//,1X,2A)
          Istop = 1
          Aline = ' '
          Do 10 I = 1,ItrLim
            If (ItrTrk(Itype,I).GT.0.0D+00) THEN
              If (Istop + 20.GE.132) THEN
                Write(12,1005) Aline(1:Istop)
 1005             Format(1X,A)
                Istop = 1
              ENDIF
              Istop = Istop + 20
              Istart = Istop - 19
              Aline(Istart:Istop) = ' (.. = .....%)  '
              Write(Aline(Istart+2:Istart+3),1010) I
```

```
 1010         Format(I2)
             Write(Aline(Istart-7:Istart+15),1015)
     &                        100.0*ItrTrk(Itype,I)/Irepet
 1015         Format(F9.5)
           ENDIF
   10   Continue
        If (Istop.GT.1) Write(12,1005) Aline(1:Istop)
   20 If (ItrTrk(Itype,0).GT.0) Write(12,1020) 100.0*ItrTrk(Itype,0)/
     &       Irepet
 1020   Format(' Iterative Failure Occurred %=',G12.5)
      Return
      END
      Double Precision Function Cslope(RlHead)
C-------------------------------------------------------------------------------------------------------------------------C
C This routine determines the water capacity of a soil column for multiple layers using 1 of 2 theoretical fits
C  To prevent division by zero Errors in certain circumstances, the lowest value
C  for water capacity that is allowed is set within a menu as CapMin. Heads are assumed to be negative values and
C capacities (slopes) are returned as positive values which cannot be less than the CapMin.
C note - the equations are using positive heads - this reverses the sign of the
C capacity - to turn it to a positive value, a minus sign has been dropped as
C the initial operator in both equations.
C-------------------------------------------------------------------------------------------------------------------------C
      INCLUDE  'Twodswap.cmb'
      HeadX = DABS(RlHead)
      IF (RlHead.GT.0.0) HeadX = 0.0D+00
      ExpM = (1.0D+00-1.0D+00/Table(3))                          (>set up exponential Muelem's model, then...
      RecpM = 1.0D+00/ExpM
      RelWat = Table(1) - Table(5)
      CapT = (1.0D+00+ (Table(4)*HeadX) ** Table(3)) **(-ExpM)   (>to the head at that node , Then calculate
      CapM = CapT**RecpM
      CtempV = Table(4) * ExpM * RelWat * CapM *                 (>finally apply the derivative of the head/theta
     &          (1.0D+00-CapM)**ExpM * (1.0D+00-ExpM)**(-1.0D+00)   (>relationship in the Van Genuchten's model
      If (CtempV.LT.CapMin) CtempV = Capmin
      Cslope = CtempV
      Return
      END
      Double Precision Function ConDuc(RlHead)
C-------------------------------------------------------------------------------------------------------------------------C
C This routine calculates ConDuctivity based on head.
C-------------------------------------------------------------------------------------------------------------------------C
      INCLUDE  'Twodswap.cmb'
      If (RlHead.LT.0.0D+00) THEN
        HeadX = DABS(RlHead)
      ELSE
        HeadX = 0.0D+00
      ENDIF
      ExpM = -1.0D+00*(1.0D+00 - 1.0D+00/Table(3))
      AlpTrm = Table(4) * HeadX
      AlpExp = 1.0D+00 + AlpTrm ** Table(3)
      CondX = Table(2) * AlpExp ** (ExpM/2.0D+00) *
     &  (1.0D+00- AlpTrm**(Table(3)-1.0D+00)*AlpExp**ExpM)**2.0D+00
      ConDuc = CondX
      Return
      END
      Double Precision Function Theta(RlHead)
C-------------------------------------------------------------------------------------------------------------------------C
C This routine calculates theta based on head.
C-------------------------------------------------------------------------------------------------------------------------C
      INCLUDE  'Twodswap.cmb'
      If (RlHead.LT.0.0D+00) THEN
        HeadX = DABS(RlHead)
        ExpM = -1.0D+00*(1.0D+00 - 1.0D+00/Table(3))
        AlpTrm = Table(4) * HeadX
        AlpExp = 1.0D+00 + AlpTrm ** Table(3)
        ThetaX = Table(5) + (Table(1) - Table(5)) * AlpExp ** ExpM
        If (Rlabel(1:12).EQ.'LINEAR THETA') THEN
          ThetaX = ThetaS*(1.0D+00-HeadX/100.0D+00)
          If (ThetaX.LT.0.08D+00) ThetaX = 0.08D+00
        ENDIF
      ELSE
        ThetaX = ThetaS
      ENDIF
      Theta = ThetaX
      Return
      END
      SUBROUTINE GtWTab(Finput)
C-------------------------------------------------------------------------------------------------------------------------
C This routine reads soil characteristics as:
C    Van Genuchten Fitted Function in Use:  (5 values)      Tsat,Ksat,N,Alpha,Est.Res.Wat
C-------------------------------------------------------------------------------------------------------------------------
      INCLUDE  'Twodswap.cmb'
      CHaracter Finput*15,Dummy*132
      Logical Error
      Call OpnFil(Finput,15,'OLD',Error)
      If (Error) Call ENdPgm('Unable to open water characteristics')
      Read(15,1000,END=20,Err=20) Dummy
 1000  Format(A)
      Read(15,*,END=20,Err=20) MWaTab
      If (MWaTab.EQ.5) THEN
        Read(15,1000,END=20) Dummy
        Read(15,*,END=20,Err=20) (Table(I),I=1,5)
        Read(15,*,END=20,Err=20) BulkDn,Dispr0,Dispr1
   15   CLose(15)
        Write(12,*)'  ---Using Van Genuchten''s Model---'
        Write(12,1010)(Table(I),I=1,5),BulkDn
 1010       Format(5X,'Tsat=',G12.5,' Ksat=',G12.5,' N=',G12.5,
     &       ' Alpha=',G12.5,' WCR=',G12.5,' BulkDn=',G12.5)
        ThetaS = Table(1)
        ConSat = Table(2)
        Return
      ENDIF
```

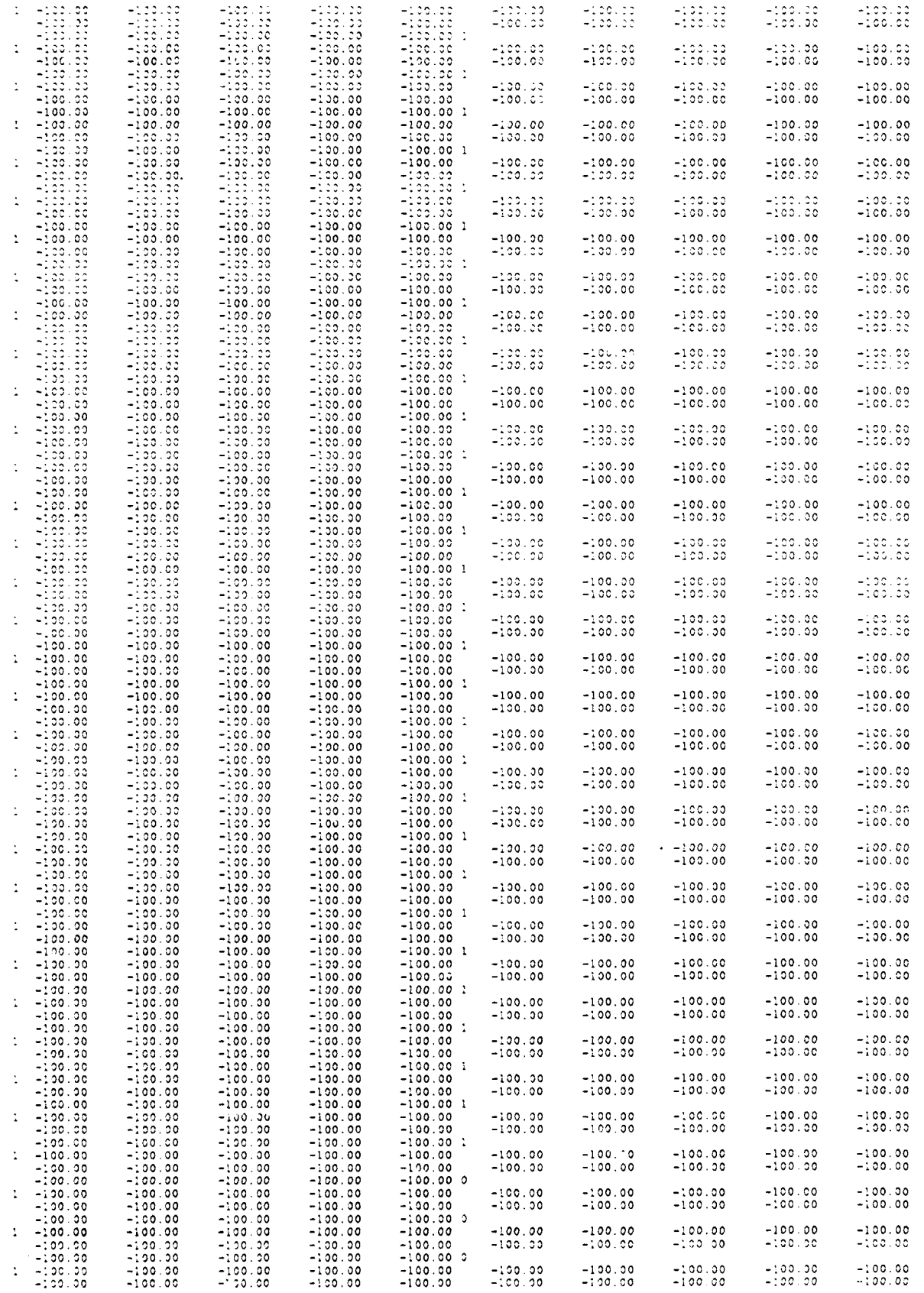
```
      Call EndPgm('Water Data:Van Gen. or cubic Function Only!')
   20 Call EndPgm('Problem with water characteristic table: '//Finput)
      END
      SUBROUTINE BrkOut
C-----------------------------------------------------------------------------------------------------------------------
C This routine appends data to the breakthrough file.
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE  'Twodswap.cmb'
      Logical InDran
      If (FBrkth.NE.' ') THEN                                    {if the BTC file is open (--> drain exists)}
        Cout = 0.0D+00
        Nout = 0
        DO 10 Row = DranMn,DranMx                                {determine concentration at the drain}
          If (OpenDr(Row)) THEN
            Nout = Nout + 1
            Cout = Cout +(Cdissv(2,Ncols,Row) + Cdissv(1,Ncols,Row))/
     &                  2.0D+00
          ENDIF
   10   Continue
        If (Nout.GT.0) THEN
          Cout = Cout/Float(Nout)
        ELSE
          Cout = 0.0D+00
        ENDIF
        Write(13,1400) Time,FlowOu(0),Cout                       {report cumulative water & current avr conc.}
 1400     Format(7G12.5)
        IBrk = IBrk + 1                                          {and increment the breakthru counter}
      ENDIF
      Return
      END
      SUBROUTINE BrkRep
C-----------------------------------------------------------------------------------------------------------------------
C This routine reads the backup Breakthrough file and converts the data into a plot data file
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE  'Twodswap.cmb'
      DIMENSION TMOUT(201),BrkTHR(0:1,201)
      Common/BrkBlk/Xout(201),Yout(201),Nobs
      Character Label*8,Rest*65
      If (FBrkth.NE.' ') THEN
        Rewind(13)
        IBrk = 0                                                 {read the data until EOF is encountered}
   10   IBrk = IBrk + 1
          If (IBrk.LE.201) THEN
            Read(13,*,END=20) TmOut(IBrk),(BrkThr(I,IBrk),I=0,1)
 1000         Format(3G12.5)
            Goto 10
          ENDIF
   20   IBrk = IBrk - 1
        Rewind(13)
        Rest = Rlabel(1:65)
        Do 30 Ispp = 0,1
          Do 25 I = 1,IBrk
            Xout(I) = TmOut(I)
   25       Yout(I) = BrkThr(Ispp,I)
          Call TrmDat                                            {reduce the data to non-repeating Y values}
          Label = Solute(Ispp)(1:8)
   30     Call Writit(Label,Rest)                                {and record it}
        ENDFILE(13)
        Close(13)
      ENDIF
      Return
      END
      SUBROUTINE TrmDat
C-----------------------------------------------------------------------------------------------------------------------
C This routine removes redunant Y axis values
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE  'Twodswap.cmb'
      Common/BrkBlk/Xout(201),Yout(201),Nobs
      Do 10 I = 2,IBrk-1                                         {if the value is the same as the adjacent values}
   10   If (DABS(Yout(I)-Yout(I-1)).LE.1.0E-06.AND.              {flag the X-value for eventual deletion of the pair}
     &      DABS(Yout(I)-Yout(I+1)).LE.1.0E-06) Xout(I) = -1.0E+30
      Nobs = 0
      Do 20 I = 1,IBrk                                           {scan the data, dropping flagged pairs}
        If (Xout(I).GE.0.0) THEN
          Nobs = Nobs + 1
          Xout(Nobs) = Xout(I)
          Yout(Nobs) = Yout(I)
        ENDIF
   20 Continue
      Return
      END
      SUBROUTINE Writit(Label,Rest)
C-----------------------------------------------------------------------------------------------------------------------
C This routine records a trimed data set to the breakthrough file in a standard plot format
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE  'Twodswap.cmb'
      Character Label*8,Rest*65
      Common/BrkBlk/Xout(201),Yout(201),Nobs
      Write(13,1000) Label,Nobs,Rest                             {Write the header, then the data}
 1000   Format(A,I5,1X,A)
      Write(13,1010) (Xout(I),Yout(I),I=1,Nobs)
 1010   Format(2G12.5)
      Return
      END
      DOUBLE PRECISION FUNCTION Retard(Icol,Irow)
C-----------------------------------------------------------------------------------------------------------------------
C This routine calculates the Retardation factor
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE  'TwoDswap.cmb'
      If (ExpLin.LE.0.1) THEN
        Write(*,9000) ExpLin
```

```
 9000     Format(' Nonlinear Sorption Exponent=',G12.5,' Reset to 1.0.')
        ExpLin = 1.0D+00
      ENDIF
      If (ExpLin.NE.1.0D+00) THEN
        AvConc = (Cdissv(1,Icol,Irow)+Cdissv(2,Icol,Irow))/2.0D+00
        SolVal = AvConc**(ExpLin-1.0D+00)
      ELSE
        SolVal =1.0D+00
      ENDIF
      Retard = 1.0D+00 + BulkDn*SlSrbC*ExpLin*SolVal/
     &                                Theta(Head(2,Icol,Irow))
      Return
      END
      Subroutine FxBndH
C-----------------------------------------------------------------------------------------------------------------------
C This routine sets the imaginary nodes appropriately for heads
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE  'Twodswap.cmb'
      Do 10 Row = 1,Nrows                                    !for ALL rows, reflect the inner nodes to the imaginary!
        Head(3,0,Row) = Head(3,1,Row)                        !nodes regardless of seep or drain designations!
   10   Head(3,Ncols+1,Row) = Head(3,Ncols,Row)
      If (SeepMx.GT.0) THEN                                  !if there is a seep!
        DO 15 Row = SeepMn,SeepMx
   15     Head(3,0,Row) = Head(3,1,Row)
      ENDIF
      If (DranMx.GT.0) THEN
        Do 20 Row = DranMn,DranMx
          If (OpenDr(Row)) Head(3,Ncols  ,Row) = 0.0D+00
   20     If (OpenDr(Row)) Head(3,Ncols+1,Row) = 0.0D+00    !if there is a drain & it's open, n=0
      ENDIF
      Do 25 Col = 1,Ncols                                    !for all columns!
        If (Reflct) THEN
          Head(3,Col,0) = Head(3,Col,1) - Dx*GravFc
        ELSE
          Head(3,Col,0) = Head(3,Col,1)
        ENDIF
   25   Head(3,Col,Nrows+1)=Head(3,Col,Nrows) + Dx*GravFc     !and the bottom is always reflective!
     &
      Head(3,0,0) = Head(3,1,0)                              !let the corners reflect the adjacent nodes
      Head(3,Ncols+1,0) = Head(3,Ncols,0)                    !these values are not used anywhere in the system!
      Head(3,0,Nrows+1) = Head(3,1,Nrows+1)
      Head(3,Ncols+1,Nrows+1) = Head(3,Ncols,Nrows+1)
      Return
      END
      Subroutine FxBndS
C-----------------------------------------------------------------------------------------------------------------------
C This routine sets the imaginary nodes appropriately for solute
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE  'Twodswap.cmb'
      Do 10 Row = 1,Nrows                                    !for ALL rows, reflect the inner nodes !
        Cdissv(2,0,Row) = Cdissv(2,1,Row)                    !regardless of seep designation!
   10   Cdissv(2,Ncols+1,Row) = Cdissv(2,Ncols,Row)         !the drain always reflects the solute!
      If (SeepMx.GT.0) THEN
        Do 15 Row = SeepMn,SeepMx
          Do 15 Jtime = 1,2
   15       Cdissv(Jtime,0,Row) = SeepC(Iflxpt)             !if there is a seep, it is always the imposed conc.!
      ENDIF
      Do 20 Col = 1,Ncols                                    !for all columns!
        If (Reflct) THEN
          Cdissv(2,Col,0) = Cdissv(2,Col,1)
        ELSE
          Cdissv(2,Col,0) = SurfC(Nzone(Col),Iflxpt)
        ENDIF
   20   Cdissv(2,Col,Nrows+1) = Cdissv(2,Col,Nrows)           !and the bottom is always reflective!
      Cdissv(2,0,0) = Cdissv(2,1,0)                          !reflect the corners though these values are not!
      Cdissv(2,Ncols+1,0) = Cdissv(2,Ncols,0)                !used!
      Cdissv(2,0,Nrows+1) = Cdissv(2,1,Nrows+1)
      Cdissv(2,Ncols+1,Nrows+1) = Cdissv(2,Ncols,Nrows+1)
      Return
      END
      Logical Function InSeep(IRow)
C-----------------------------------------------------------------------------------------------------------------------
C This routine determines if a node is in the seep and if the seep is active
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE  'Twodswap.cmb'
      InSeep = .false.
      If (SeepMx.GT.0.AND.Irow.GE.SeepMn.AND.Irow.LE.SeepMx)
     &                                        InSeep = .true.
      RETURN
      END
      Logical Function InDran(Irow)
C-----------------------------------------------------------------------------------------------------------------------
C This routine determines if a given rwo is within the drain
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE  'Twodswap.cmb'
      InDran = .false.
      If (DranMx.GT.0.AND.Irow.GE.DranMn.AND.Irow.LE.DranMx)
     &                                        InDran = .true.
      RETURN
      END
      SUBRoutine StartR
C-----------------------------------------------------------------------------------------------------------------------
C This routine starts the simulation run by initialization & reading the parameter file
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE  'Twodswap.cmb'
      Character Fparam*15
      Data Irepet/0/,TmLast/0.0/,TmSpan/0.0/,Time/0.0D+00/,Ptime/0/,
     &     DtStar/.true./,WatFix/.false./Iflxpt/0/,FlowIn/3*0.0/,
     &     Solute/'Water','Solute'/,Nstable/0/,FlowOu/3*0.0/,
     &     SolFix/.false./,Reflct/.true./
      write(*,1000)
```

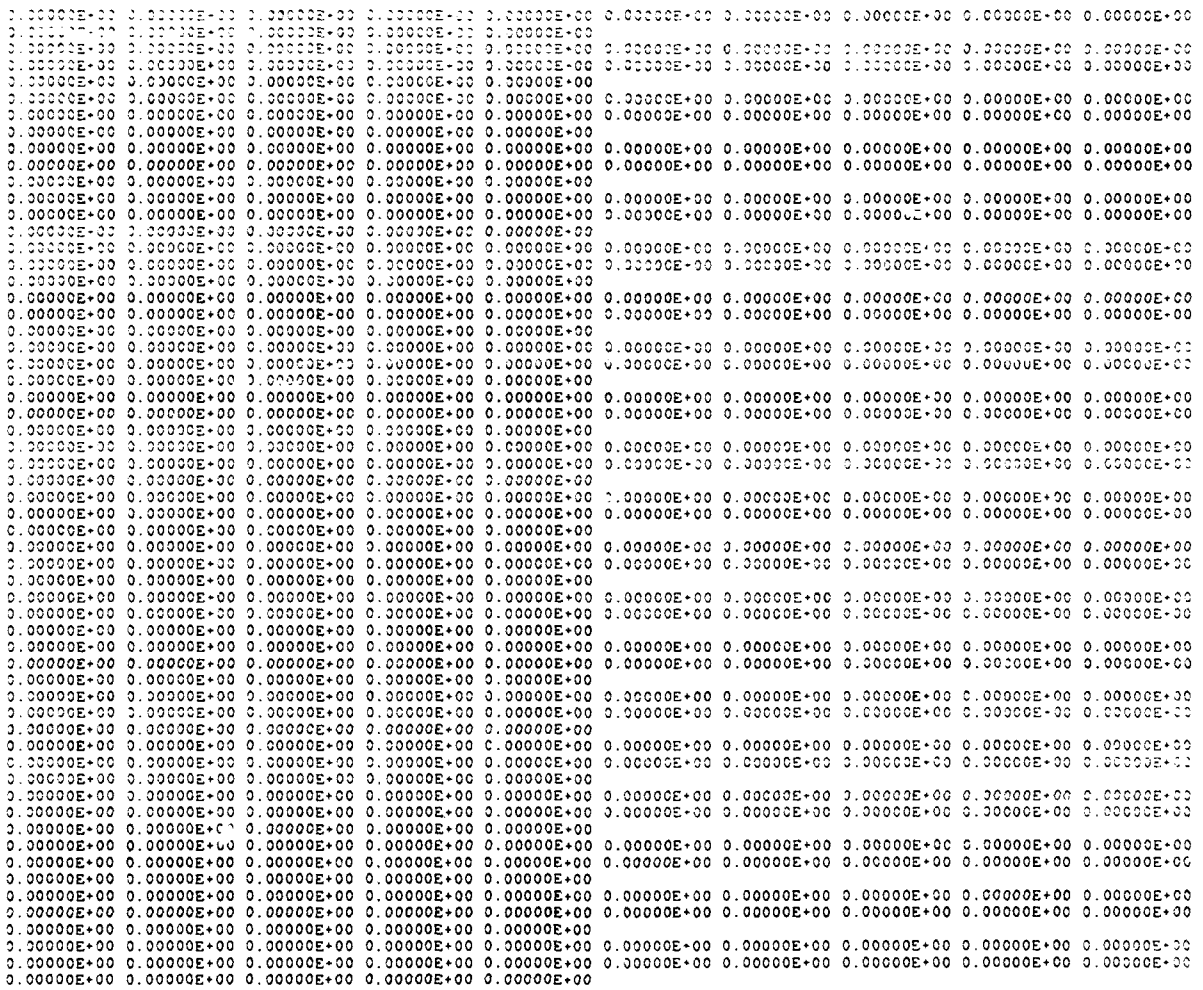
```
 1000   Format('$Enter Parameter FileName: ')
      Read(*,1010) Fparam
 1010   Format(A)
      Open(Unit=10,File=Fparam,Status='OLD')
      Call GetPar                                                    {read the run parameters}
      If (DranMx.GT.0) THEN
        Do 15 Row = DranMn,DranMx
          If (Head(1,Ncols,Row).GE.0.0) THEN
            DO 10 Itime = 1,3                                        {the heads at the drain to 0 (at all times) &}
              DO 10 Col = Ncols,Ncols+1
   10           Head(Itime,Col,Row) = 0.0D+00
            OpenDr(Row) = .true.
          ELSE
            OpenDr(Row) = .false.
          ENDIF
   15   Continue
      ENDIF
      Return
      END
      SUBROUTINE GetPar
C-----------------------------------------------------------------------------------------------------------------------
C This routine reads the start parameter file
C-----------------------------------------------------------------------------------------------------------------------
      INCLUDE  'Twodswap.cmb'
      Logical FMopen,Error,AskQus,Abort
      CHARACTER Ftable*15,Lo2Chr*1
      SolMax = 0.0D+00
      Read(10,1000) Fprint                                           {read the file name for the output & opens that file}
 1000     Format(A)
        Call OpnFil(Fprint,12,'NEW',Error)
        Call StrCap(Fprint)
        If (Error) Call ENdPgm('Unable to open results file')
      Write(12,1005) Char(12)                                        {Write a simulation Report heading}
 1005  Format(A,32X,'Two-Dimensional Transient Flow - 080988 - ',
     &   'DR.S.A.BLOOM',/,
     &  ' ',128('-'),'+',/,28X,'SIMULATION PARAMETERS:',/)
      Write(12,1000) ' '//Fprint                                     {now read & echo the various parameters}
      Call GetStr(ABort,Rlabel)                                      {the run label}
      Call GetStr(ABort,Ftable)                                      {the head/solute boundary table file-name}
      Call GtHTab(Ftable)                                            {now read that information}
      If (DranMx.GT.0) THEN                                          {if there is a drain, open the BTC file}
        Fbrkth = Fprint(1:Iextnt(Fprint))//'BRK'
        Call OpnFil(Fbrkth,13,'NEW',Error)
        If (Error) Call ENdPgm('Unable to open BTC file')
      ELSE
        Fbrkth = ' '
      ENDIF
      Call GtReal(Abort,TmStop)                                      {the time to stop the simulation}
      Call GtReal(Abort,CapMin)                                      {the minimum water capacity (>0)}
      Call GtReal(Abort,GravFc)                                      {the gravity factor -> 1 is vertical}
      Call GtReal(Abort,SlSrbC)                                      {the sorption coefficient - controls retardation}
      ExpLin = 1.0                                                   {nonlinearity retardation coefficient default=1}
      If (DABS(SlSrbC).GT.1.0E-06) Call GtReal(Abort,ExpLin)         {if retardation is wanted, then read the exponent}
      Call GtReal(Abort,Dt)                                          {maximum Dt}
      DtMin = 0.1
      DtMax = Dt
      Call GetStr(ABort,Ftable)                                      {soil/water characteristics file-name & read it}
      Call GtWTab(Ftable)
 1010   Format(A,G12.5,A,G12.5)
      Call GetInt(Abort,Nflux)                                       {the number of flux changes during the run}
      If (Nflux.GT.Maxset) Call EndPgm('Too many flux sets')
      Do 20 I = 1,Nflux
        Call GtReal(Abort,FluxTm(I))                                 {the time of each flux change}
        Do 15 Iz = 1,Nzone(Ncols)
          Call GtReal(Abort,SurfQ(Iz,I))                             {read the surface flux rate (cm/hr) & convert it}
          SurfQ(Iz,I) = SurfQ(Iz,I)/3600.0
          Write(12,1015) Iz,' Surface flux-->cm/sec =',SurfQ(Iz,I)
 1015      Format(' Zone ',I2,A,G12.5)
          If (SurfQ(Iz,I).GT.ConSat) Call EndPgm('Surface FLux > Ksa.') {check for exceeding entrance capacity}
          If (Ncols*SurfQ(Iz,I).GT.(DranMx-DranMn+1)*Consat) THEN    {warn if exceeding the drainage capacitt
            Write(12,*) ' DANGER: Input will exceed drainage ability'
            Write(*,*) ' DANGER: Input will exceed drainage ability'
          ENDIF
          Call GtReal(Abort,SurfC(Iz,I))                             {concentration in the 'rain'}
          Write(12,1015) Iz,' Surface salt conc-->meq/cc =',SurfC(Iz,I)
          If (SurfC(Iz,I).GT.Solmax) SolMax = SurfC(Iz,I)
          If (SurfQ(Iz,I).LT.0.0) THEN                               {make sure that conc=0 if flux is negative}
            SurfC(Iz,I) = 0.0D+00
            Write(12,1010) ' Evaporation requires conc.to be 0'
          ENDIF
   15   Continue
        If (SeepMx.GT.0) THEN                                        {if there is a seep (horizontal input area)...}
          Call GtReal(Abort,SeepQ(I))                                {read the head or flux at that point}
          SeepQ(I) = SeepQ(I)/3600.0                                 {if flux, convert to cm/sec from cm/hr}
          If (SeepQ(I).GT.ConSat) Call EndPgm('Seepage FLux > Ksat') {and check for exceeding entrance capacity}
          Write(12,1010) ' Seepage flux--> cm/sec =',SeepQ(I)
          If (SeepQ(I).GT.ConSat) Call EndPgm('Seep FLux > Ksat')
          Call GtReal(Abort,SeepC(I))                                {concentration at the seep}
          Write(12,1010) ' Solute conc at seep-->meq/cc=',SeepC(I)
          If (SeepC(I).GT.Solmax) SolMax = SeepC(I)
        ENDIF
   20 Continue
      Call GetInt(Abort,Nrepr)                                       {number of reports requested}
      If (Nrepr.GT.Maxrep) Call EndPgm('Too many Reports requested')
      FMopen = .false.                                               {initialize reporting constraints}
      TmRepr(0) = 0.0
      RepHed(0) = .false.
      RepThe(0) = .false.
      Do 60 I = 1,Nrepr
        Call GtReal(Abort,TmRepr(I))                                 {the time of the requested report (secs)}
        Call GetLog(Abort,RepHed(I))                                 {true if the heads are to be reported}
```

```
      Call GetLog(Abort,RepThe(I))                          {true if theta is to be reported}
      Call GetLog(Abort,RepSol(I))                          {true if the concentrations are to be reported}
      Call GetLog(Abort,RpConc(I))                          {true if the absolute amounts are to be reported}
 60   If (RepHed(I).OR.RepThe(I).OR.RepSol(I).OR.RpConc(I)) {if anything is being reported, set the flag to T}
     &     FMOpen = .true.
      If (FMOpen) THEN                                      {if the flag is T, then open the matrix report file}
        Fmatrx = Fprint(1:Iextnt(Fprint))//'2DM'
        Call OpnFil(Fmatrx,14,'NEW',Error)
        If (Error) Call ENdPgm('Unable to open 2DM file')
      ENDIF
      IF (TMREPR(NREPR).LT.TmStop) THEN                     {insure last report corresponds to termination}
        NREPR = NREPR + 1
        TMREPR(NREPR) = Tmstop
        RepHed(Nrepr) = RepHed(Nrepr-1)
        RepThe(Nrepr) = RepThe(Nrepr-1)
        RepSol(Nrepr) = RepSol(Nrepr-1)
      ENDIF
      Call GetLog(Abort,Feedbk)                             {boolean controlling feedback comments for Debuging}
      Call GetInt(Abort,Icheck)                             {the frequency of checking the abort file}
      Call GtReal(Abort,Tlevel)                             {the iterative tolerance level (0.01 or 0.05 usually)}
      If (Tlevel.GT.0.0) THEN
        Strict = .true.
      ELSE
        Strict = .false.
        Tlevel = -Tlevel
      ENDIF
      Call GetInt(Abort,ItrLim)                             {the number of iterative trys before giving up}
      If (ItrLim.GT.30) Call EndPgm('Iterative Limit <=30 only')
      Call GtReal(Abort,DtStep)                             {the % Dt increment used to step Dt from DtMin to DtMax}
      If (Dtmin.EQ.DtMax.AND.Dt.EQ.DtMax) DtStep = 1.0
 65   Close(10)
      Reflct = .true.
      DO 70 Col = 1,Nzone(Ncols)
 70     If (DABS(SurfQ(Nzone(Col),Iflxpt)+SurfC(Nzone(Col),Iflxpt))
     &     .GT.1.0D-06.OR.Rlabel(1:6).NE.'REFLCT') Reflct = .false.
      Call ColSet                                           {set the head & solute values in space & time}
      Write(12,1050) Icheck,Tlevel,ItrLim,DtStep,Lo2Chr(Reflct)
1050  Format(' External abort check       = ',I12,/,
     &       ' Iterative Tolerance Limit = ',G12.5,' ( %)',/,
     &       ' Iterative Pass Maximum    = ',I12,/,
     &       ' Step Amount for Dt change = ',G12.5,' (frac)',/
     &       ' Top boundary a reflection = ',A)
      Return
      END
```

```
Parameter file for the Two-Dimensional Transient Program

SurSpill.Res    : Filename for results storage
Central spill (2cm/hr @ 100 solute) with gravity
Surspill.mnd     : file with initial head/dx/nrows/ncol info
  18000.0        : Time to terminate simulation (sec)
 3.0E-05         : Minimum Water Capacity
  1.0            : Gravitational Factor
  0.0            : Solute Sorption Factor
  100.0          : Dt Max
livoakup.vag     : Van Genuchten Model Coefficients for Water Characteristics
               1 : Number of flux changes during run
  0.0            : Time (sec) of flux change
  3.0            : Surface flux (cm/hr)
 100.0           : Solute concentration
  0.0            : Surface flux (cm/hr)
  0.0            : Solute concentration
               4 : Number of reports to be generated
   1000.0        : Report to be generated (sec)
             Yes : Report the head matrix
              No : Report the Theta matrix
             Yes : Report the solute Concentration matrix
              No : Report the Absolute  solute matrix
   6000.0        : Report to be generated (sec)
             Yes : Report the head matrix
              No : Report the Theta matrix
             Yes : Report the solute Concentration matrix
              No : Report the Absolute  solute matrix
  12000.0        : Report to be generated (sec)
             Yes : Report the head matrix
              No : Report the Theta matrix
             Yes : Report the solute Concentration matrix
              No : Report the Absolute  solute matrix
  18000.0        : Report to be generated (sec)
             Yes : Report the head matrix
              No : Report the Theta matrix
             Yes : Report the solute Concentration matrix
              No : Report the Absolute  solute matrix
              No : Feedback for debugging
              20 : abortion check periodicity
  0.5            : Tolerance Limit in percentage
              20 : Number of passes before iterative failure
  0.05           : Increment for Dt - fractional

Auxillary Parameter file for the Two-Dimensional Transient Program   ---Water Characteristics File---

Lakeland - Live Oak Sand (S-124) - Upper 30 cm (Van Genuschten only!)
  5  0.0000      1.0000     = Numbers of rows for Function VG. fitting
 Theta Sat  KSat(cm/sec)      N        Est. ALpha    Residual Wat.
0.35200000 0.34310999E-02 4.1901102 0.20310000E-01  0.85770003E-01
  1.65     0.43E-07 1.0 : Bulk Density (gm/cc)
  1.65     0.000278 1.0  : Bulk Density (gm/cc)

Auxillary Parameter file for the Two-Dimensional Transient Program   ---Two-Dimensional Boundary Condition File---


Corner Spill
   25   50  1.0000
        1           1           1           1           1           2           2           2           2           2
     2           2           2           2           2           2           2           2           2           2
     2           2           2           2           2
 1  -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00
    -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00
    -100.00     -100.00     -100.00     -100.00     -100.00 1
 1  -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00
    -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00
    -100.00     -100.00     -100.00     -100.00     -100.00 1
 1  -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00
    -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00
    -100.00     -100.00     -100.00     -100.00     -100.00 1
 1  -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00
    -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00
    -100.00     -100.00     -100.00     -100.00     -100.00 1
 1  -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00
    -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00
    -100.00     -100.00     -100.00     -100.00     -100.00 1
 1  -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00
    -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00
    -100.00     -100.00     -100.00     -100.00     -100.00 1
 1  -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00
    -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00
    -100.00     -100.00     -100.00     -100.00     -100.00 1
 1  -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00
    -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00
    -100.00     -100.00     -100.00     -100.00     -100.00 1
 1  -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00
    -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00
    -100.00     -100.00     -100.00     -100.00     -100.00 1
 1  -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00
    -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00
    -100.00     -100.00     -100.00     -100.00     -100.00 1
 1  -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00
    -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00
    -100.00     -100.00     -100.00     -100.00     -100.00 1
 1  -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00
    -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00
    -100.00     -100.00     -100.00     -100.00     -100.00 1
 1  -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00
    -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00     -100.00
    -100.00     -100.00     -100.00     -100.00     -100.00 1
```



```
-100.00 -100.00 -100.00 -100.00 -100.00 0
-100.00 -100.00 -100.00 -100.00 -100.00 -100.00 -100.00 -100.00 -100.00 -100.00
-100.00 -100.00 -100.00 -100.00 -100.00 -100.00 -100.00 -100.00 -100.00 -100.00
-100.00 -100.00 -100.00 -100.00 -100.00 0
-100.00 -100.00 -100.00 -100.00 -100.00 -100.00 -100.00 -100.00 -100.00 -100.00
-100.00 -100.00 -100.00 -100.00 -100.00 -100.00 -100.00 -100.00 -100.00 -100.00
-100.00 -100.00 -100.00 -100.00 -100.00 0
-100.00 -100.00 -100.00 -100.00 -100.00 -100.00 -100.00 -100.00 -100.00 -100.00
-100.00 -100.00 -100.00 -100.00 -100.00 -100.00 -100.00 -100.00 -100.00 -100.00
-100.00 -100.00 -100.00 -100.00 -100.00 1
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
```



```
C. Utility subroutines used in the preparation and simulation programs

      LOGICAL Function AskQus(Query,Clear,Line)                         (      SUBROUTINE)
C-----------------------------------------------------------------------------------------------------------------------------
C This Function returns as either true or false - it displays the query centered on LINE and will clear the screen if CLEAR
C is true
C-----------------------------------------------------------------------------------------------------------------------------
C
      Character*(*) Query,Blank*80,Answer*10,Questn*132
      Logical Clear,NullOK
      Data Blank/' '/
      Call StrLen(Query,Length)
      AskQus = .false.
      If (Query(Length:Length).EQ.Char(8)) THEN
         Questn = '\'//Query
         NullOk = .true.
         If (Query(Length-1:Length-1).EQ.'Y') AskQus = .true.
      ELSE
         Questn = '\'//Query//' (Y/N): '
         NullOk = .false.
      ENDIF
   30 Call Notice(Questn,Clear,Line)
      Read(*,1000,Err=30) Answer
 1000   Format(A)
      Call ShfTxt(Answer)
      Call StrLen(Answer,LenAns)
      If (.not.NullOk.AND.LenAns.LE.0) Goto 30
      If (.Not.NullOk.OR.LenAns.GT.0) THEN
         If (Answer(1:1).EQ.'Y'.OR.Answer(1:1).EQ.'y') THEN
            AskQus = .True.
         ELSE If (Answer(1:1).EQ.'N'.OR.Answer(1:1).EQ.'n') THEN
            AskQus = .false.
         ELSE
            Goto 30
         ENDIF
      ENDIF
      Return
      END
      Logical Function CharOK(Letter)
C-----------------------------------------------------------------------------------------------------------------------------
C This routine checks the characters in a proposed file name
C-----------------------------------------------------------------------------------------------------------------------------
      CHaracter Letter*1
      If ( (Letter.GE.'A'.AND.Letter.LE.'Z').OR.
     &       (Letter.GE.'0'.AND.Letter.LE.'9').OR.
     &        Letter.EQ.':') THEN
        CharOk = .true.
      ELSE
        CharOk = .false.
      ENDIF
      Return
      END
      Subroutine EchoLo(Bolean)
C-----------------------------------------------------------------------------------------------------------------------------
C This routine reads and echos a boolean setting from the parameter file
C-----------------------------------------------------------------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      Character ALine*80,YesNo*3
      Logical Bolean
      Read(10,1000) YesNo,ALine
 1000   Format(12X,2A)
        Write(12,1000) YesNo,ALine(1:LenOrl(Aline))
        Bolean = Str2Lo(YesNo)
      Return
      END
      Subroutine EchoRl(Avalue)
C-----------------------------------------------------------------------------------------------------------------------------
C This routine reads and echos a real setting from the parameter file
C-----------------------------------------------------------------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      Character ALine*80
      Read(10,1000) Avalue,ALine
 1000   Format(G15.7,A)
        Write(12,1000) Avalue,ALine(1:LenOrl(Aline))
      Return
      END
      Subroutine EchoIn(Ivalue)
C-----------------------------------------------------------------------------------------------------------------------------
C This routine reads and echos an integer setting from the parameter file
C-----------------------------------------------------------------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      Character ALine*80
      Read(10,1000) Ivalue,ALine
 1000    Format(I15,A)
        Write(12,1000) Ivalue,ALine(1:LenOrl(ALine))
      Return
      END
      Subroutine EchoSt(Text)
C-----------------------------------------------------------------------------------------------------------------------------
C This routine reads and echos a string setting from the parameter file
C-----------------------------------------------------------------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      Character*(*) Text,ALine*80
      LenTxt = Len(Text)
      Read(10,1000) ALine
 1000  Format(A)
        Write(12,1000) ALine(1:LenOrl(ALine))
        Text = ALine(1:LenTxt)
      Return
```

```
      END
      SUBROUTINE GTFLNM(QUERY,FNAME)
C-----------------------------------------------------------------------------------------------------------------------
C        prompts the user with QUERY to supply a file name (FNAME)
C-----------------------------------------------------------------------------------------------------------------------
C
      CHARACTER*(*) Fname,Query,Ftemp*50,Qtemp*132
      Logical AskQus
      Qtemp = Query
      CALL STRLEN(Qtemp,LENGTH)
   10 Call Notice('NOTE: There is no validity Check on File Names',
     &                    .true.,6)
   11 CALL Notice('\\Please enter filename for '//Qtemp(:LENGTH)//
     &                    ': ',.false.,10)
      READ(*,1010,Err=11) FNAME
 1010  FORMAT(A)
      Ftemp = Fname
      Call StrLen(Ftemp,LongFl)
      If (LongFl.LE.0) Goto 11
      Call ShfTxt(Ftemp)
      Call StrCap(Ftemp)
      Call StrLen(Ftemp,LongFl)
      If (.Not.AskQus('\Filename for '//Qtemp(:LENGTH)//' is '//
     &     Ftemp(:LONGFL)//'? (Y/N): Y'//Char(8),.false.,14)) Goto 11
      Fname = Ftemp(1:Len(Fname))
      RETURN
      END
      SUBROUTINE In2Chr(IValue,String)
C-----------------------------------------------------------------------------------------------------------------------
C this routine converts a boolean into a character representation
C-----------------------------------------------------------------------------------------------------------------------
      Character*(*) String
      Length = Len(String)
      String = ' '
      IR = IABS(IValue)
      IF (IR.EQ.0) THEN
        String(Length:Length) = '0'
      ELSE
        Nplace = INT(ALog10(Float(IR)))
        If ((Length.LT.Nplace  .AND.Ivalue.GT.0).OR.
     &      (Length.LT.Nplace+1.AND.Ivalue.LT.0)    ) THEN
          Do 10 I = 1,Length
   10       String(I:I) = '*'
        ELSE
          Iptr = Length
   20     String(Iptr:Iptr) = CHar(Ichar('0')+MOD(IR,10))
          IR = IR/10
          Iptr = Iptr - 1
          If (Iptr.GE.1.AND.IR.GT.0) Goto 20
          If (IValue.LT.0) String(Iptr:Iptr) = '-'
        ENDIF
      ENDIF
      Return
      END
      SUBRoutine Int2Ch(String,Ivalue,Length)
C-----------------------------------------------------------------------------------------------------------------------
C This routine converts an integer into a string variable
C-----------------------------------------------------------------------------------------------------------------------
C
      Character*(*) String
      String = ' '
      If (Ivalue.Eq.0) THEN
        String(1:1) = '0'
      ELSE
        Length = Len(String)
        Nchar = INT(Alog10(Float(IABS(Ivalue)))+1)
        If (Ivalue.LT.0) Nchar = Nchar + 1
        If (Nchar.GT.Length) THEN
          Do 10 I = 1,Length
   10       String(I:I) = '*'
        ELSE
          Left = Ivalue
          Do 20 I = 1,Nchar
            String(Nchar+1-I:Nchar+1-I)=Char(Ichar('0')+Mod(Left,10))
   20       Left = Left/10
          If (Ivalue.LT.0) String(1:1) = '-'
        ENDIF
      ENDIF
      Call StrLen(String,Length)
      Return
      END
      Integer Function Lenorl(Text)
C-----------------------------------------------------------------------------------------------------------------------
C-----------------------------------------------------------------------------------------------------------------------
      Character*(*) Text
      Call StrLen(Text,Length)
      If (Length.LE.0) Length = 1
      LenOrl = Length
      Return
      END
      Real Function Lo2Rel(Boolen)
C-----------------------------------------------------------------------------------------------------------------------
C-----------------------------------------------------------------------------------------------------------------------
      Logical Boolen
      If (Boolen) THEN
        Lo2Rel = 1.0
      ELSE
        Lo2Rel = 2.0
      ENDIF
      Return
      END
```

```
      INTEGER Function MenChc(LenLin,Nitems,Option,Alter)              (      SUBROUTINE Flag)
C-------------------------------------------------------------------------------------------------------------------------------
C This Function displays a static choice menu and returns an integer value Minchc<=value<=MaxChc. NOTE that Option must be an
C 80-character string.
C-------------------------------------------------------------------------------------------------------------------------------
      Character Option(*)*80,Blanks*80
      Logical Alter
      Blanks = ' '
      MinChc = 25
      MaxChc = 0
      LinSkp = (15 - Nitems) / Nitems                                  (cal.the number of lines to be skipped between items)
      Call Page('Dummy')
      Call Vtab(3)
      If (alter) Write(*,1000)
 1000   Format(18X,'Please enter the number of your selection')
      Call Vtab(5)                                                      (starting at line 5)
      Margin = (80-LenLin)/2.0
      If (Margin.LE.0) Margin = 1
      Do 20 Item = 1,Nitems
         Call StrLen(Option(Item),Length)
         If (Option(Item)(1:1).EQ.'<') THEN
            Nvalue = Ichar(Option(Item)(2:2))-Ichar('0')
            If (Option(Item)(2:2).EQ.' ') Nvalue = 0
            If (Option(Item)(3:3).NE.'>') Nvalue = Nvalue*10 +
     &         Ichar(Option(Item)(3:3))-Ichar('0')
            If (Nvalue.LT.MinChc) MinChc = Nvalue
            If (Nvalue.GT.MaxChc) MaxChc = Nvalue
         ENDIF
         If (Length.GT.0) THEN
            If (Option(Item)(1:1).EQ.'<') THEN
               Lstop = Margin
            ELSE
               Lstop = NINT( (80-Length)/2.0 )
            ENDIF
            Write(*,1010) Blanks(1:Lstop),Option(Item)(1:Length)
 1010          Format(2A)
         ELSE
            Write(*,1010) ' '
         ENDIF
         If (LinSkp.GT.0) THEN
           Do 19 I = 1,LinSkp
   19        Write(*,1010) ' '
         ENDIF
   20 Continue
      If (Minchc.LE.MaxChc.AND.MinChc.GE.0.AND.Alter) THEN
         MenChc = MnuChc(MinChc,MaxChc)
      ELSE
         MenChc = 0
      ENDIF
      Return
      END
C
      SUBROUTINE MenFil(LenQry,NQuery,Query,Answer,String,Alter)
C-------------------------------------------------------------------------------------------------------------------------------
C This subroutine displays an interactive (user supplied values and/or agree to displayed default value) menu. It first
C displays the menu as provided by the calling routine (see PROCQU comments for proper encoding pattern) and then allows the
C user to alter items (with the exception of comment lines and item numbers that do no exist) within the range of items
C numbers between 1 and MaxVal. When '0' is chosen, the routine returns control to the calling routine.
C Note that Query and String must both be 80 character strings.  The variable LENQRY controls the number of characters to be
C printed (8:LenQry) except for Queries with the 'S' in Column 5. In those cases, the expected length of STRING must be placed
C in ANSWER, i.e. if the 3rd Query asked for an 8-char. answer, then ANSWER(3) = 8. The routine will determine whether the
C minimum query or LenQry characters will be displaced by checking the length of the query, LenQry and the expected STRING
C length.  The maximum possible menu choice is determined from the query arguments by the subroutine.
C-------------------------------------------------------------------------------------------------------------------------------
C
      DImension Answer(*)
      Character Query(*)*80,String(*)*80
      Integer Choice
      Logical Alter
      MaxVal = 0
    1 Call Page('DUmmy')
      Call Vtab(3)
      If (Alter) Write(*,1000)'        Please enter...            '
      Do 10 I = 1,Nquery
   10   Call ProcQu(LenQry,Query(I),String(I),ANswer(I),MaxVal)
      Call Vtab(3)
      If (Alter) Write(*,1000)' < 0> Accept Answers as displayed  '
 1000   Format(A)
      If (Alter) THEN
   20   Choice = MnuChc(0,MaxVal)
        IF (Choice.GT.0) THEN
           I = 0
C     ....REPEAT....
   30         I = I + 1
              Item = (Ichar(Query(I)(3:3))-Ichar('0'))*10 +
     &               (Ichar(Query(I)(4:4))-Ichar('0'))* 1
C     ....UNTIL....
            If (Item.NE.Choice.AND.I.LT.Nquery) Goto 30
           If (Item.EQ.Choice) THEN
              Query(I)(6:6) = '?'
              Call ProcQu(LenQry,Query(I),String(I),Answer(I),MaxVal)
           ENDIF
           Goto 20
        ENDIF
      ENDIF
      Return
      END
      INTEGER Function MnuChc(Minmum,Maxmum)                           (      SUBROUTINE flag)
C-------------------------------------------------------------------------------------------------------------------------------
C This Function displays the selection prompt & returns the choice
C-------------------------------------------------------------------------------------------------------------------------------
```

```
C
      Character ValMin*2,ValMax*2,Questn*132
      Integer Choice
      Call Int2Ch(ValMin,Minmum,LenMin)
      Call Int2Ch(ValMax,Maxmum,LenMax)
      Questn = '\Your Selection ('//ValMin(1:LenMin)//' to '//
     &     ValMax(1:LenMax)//'): '
   10 Call Notice(Questn,.false.,22)
      Call Value(Respon,IER)
      Choice = NINT(Respon)
      If (Choice.LT.Minmum.OR.CHoice.GT.Maxmum.OR.IER.NE.0) Goto 10
      MnuChc = Choice
      RETURN
      END
      SUBROUTINE Notice(Messag,Clear,Line)
C-------------------------------------------------------------------------------------------------------------------------------
C This subroutine displays the MESSAge centered on LINE and will clear the screen if CLEAR  is true.
C-------------------------------------------------------------------------------------------------------------------------------
C
      Character*(*) Messag
      Character*80 Blank
      Logical Clear,Center
      Data Blank/' '/
      Center = .true.
      If (Clear) Call Page('Dummy')
      Call Strlen(Messag,Length)
      If (Length.LE.0) Length = 1
      If (Messag(Length:Length).EQ.':') Length = Length + 1
      If (Messag(1:1).EQ.'\') THEN
        If (Messag(2:2).EQ.'\') THEN
           Istart = 3
           Center = .false.
        ELSE
           Istart = 2
        ENDIF
      ELSE
        Istart = 1
      ENDIF
      LinAct = Line
      If (Length.GT.75) THEN
        Do 10 I = 1,75
            If (Messag(76-I:76-I).EQ.' ') Goto 20
   10     Continue
          I = 75
   20     Ibreak = I
          Margin = (80 - Ibreak) / 2
          If (Margin.LE.0) Margin = 1
          Call Vtab(Line)
          Write(*,1000) Blank(1:Margin),Messag(Istart:Ibreak)
C 1000        Format(2A,\) -----backslash----
 1000         Format('$',2A)
          Write(*,1010)
 1010        Format(/)
          LinAct = Line + 1
          Istart = Ibreak+1
      ENDIF
      Margin = (80 - Length+1-Istart) / 2
      If (.Not.Center.OR.Margin.LE.0) Margin = 1
      Call Vtab(LinAct)
      Write(*,1000) Blank(1:Margin),Messag(Istart:Length)
      If (Messag(1:1).NE.'\') Write(*,1010)
      Return
      END
      SUBROUTINE PAGE(PGMNAM)
C-------------------------------------------------------------------------------------------------------------------------------
C        clears the screen and displays a top title centered & in inverse Video
C-------------------------------------------------------------------------------------------------------------------------------
C
      CHARACTER*(*) PGMNAM
      CHARACTER TOPLAB*80,BLANKS*80,Home*7,Invron*4,InVrOf*4
      data Blanks/' '/,INIT/0/
      IF (PGMNAM(:5).EQ.'RESET') THEN
         INIT = 0
         RETURN
      ENDIF
      IF (INIT.EQ.0) THEN
         HOME = CHAR(27)//'[2J'//CHAR(27)//'[H'
         INVRON = CHAR(27)//'[7m'
         INVROF = CHAR(27)//'[0m'
         INIT = 1
         Call StrBeg(PgmNam,Istart)
         Call StrLen(PgmNam,Istop)
         TOPLAB = ' '//PGMNAM(Istart:IStop)//' '
         LAST = Istop - Istart + 3
         LEFMAR = INT( (80- Last )/2)
         RETURN
      ENDIF
      Write(*,1000) HOME,BLANKS(:LEFMAR),INVRON,TOPLAB(:LAST),INVROF
 1000  FORMAT(1X,5A)
      RETURN
      END
      SUBROUTINE ProcQu(LenQry,Query,String,Answer,MaxVal)
C-------------------------------------------------------------------------------------------------------------------------------
C This subroutine processes a query. Information on where to display the query, the type of answer needed and the item number
C are encoded as: Char.1 & 2 : 2-digit right-justified number which determines which line on the screen is used (01-24)
C                       3 & 4 : 2-digit right-justified number which will be used as an item number in the display
C                           5 : Variable Type : L,I,R,S,C -> logical,integer,real,character string, or comment
C                           6 : L,C,R,D,? -> Comment left,right or centered,default (value set) or need value from the user
C-------------------------------------------------------------------------------------------------------------------------------
C
      Character*(*) Query,String
```

```
      Character YesNo(2)*3,Blank*80
      Data YesNo/'Yes',' No'/,Blank/' '/
      Line = (Ichar(Query(1:1))-Ichar('0'))*10 +
     &       (Ichar(Query(2:2))-Ichar('0'))* 1
      Item = (Ichar(Query(3:3))-Ichar('0'))*10 +
     &       (Ichar(Query(4:4))-Ichar('0'))* 1
      If (Item.GT.MaxVal) MaxVal = Item
   10 Call Vtab(Line)
      IF (Query(5:5).EQ.'S') THEN                                  {Character string requested}
         Call StrLen(Query,Length)
         If (Answer+LenQry-8.LE.80) Length = LenQry
         IF (Query(6:6).EQ.'?') THEN
            Write(*,1010) Blank
            Call Vtab(Line)
            Write(*,1000) Item,Query(8:Length)
C 1000         Format(' <',I2,'> ',A,': ',\)---backslash---
 1000        Format('$<',I2,'> ',A,': ')
            Read(*,1010) String
 1010         Format(A)
         ELSE
            Call StrLen(String,LenRlb)
            If (LenRlb.LE.0) LenRlb = 1
            Write(*,1020) Item,Query(8:Length),String(1:LenRlb)
 1020         Format(' <',I2,'> ',A,': ',A)
         ENDIF
      ELSE IF (Query(5:5).EQ.'L') THEN                             {Logical variable - will output 'yes' or ' no'}
         IF (Query(6:6).EQ.'D') THEN
            Write(*,1030) Item,Query(8:LenQry),YesNo(NINT(Answer))
 1030         Format(' <',I2,'> ',A,': ',9X,A)
         ELSE
            IF (Nint(Answer).EQ.1) THEN                            {if changing setting, then if it was 1 (=Yes), switch
to
               Answer = 2.0                                        { 2 (=No) or vise versa}
            ELSE
               Answer = 1.0
            ENDIF
         ENDIF
      ELSE IF (Query(5:5).EQ.'R') THEN                             {real values}
         IF (Query(6:6).EQ.'D') THEN
            IF (Answer.GE.0.00001.AND.Answer.LE.999999.0) THEN     {filter for range and relate to format display}
               Write(*,1040) Item,Query(8:LenQry),Answer
 1040            Format(' <',I2,'> ',A,': ',F12.5)
            ELSE
               Write(*,1050) Item,Query(8:LenQry),Answer
 1050            Format(' <',I2,'> ',A,': ',E12.5)
            ENDIF
         ELSE
             Write(*,1060) Item,Query(8:LenQry)
C 1060         Format(' <',I2,'> ',A,': ',\)---backslash---
 1060        Format('$<',I2,'> ',A,': ')
            Call Value(Answer,IER)
            IF (IER.NE.0) Goto 10
         ENDIF
      ELSE IF (Query(5:5).EQ.'I') THEN                             {Integer values}
         IF (Query(6:6).EQ.'D') THEN
            Write(*,1070) Item,Query(8:LenQry),Nint(Answer)
 1070         Format(' <',I2,'> ',A,': ',I12)
         ELSE
            Write(*,1080) Item,Query(8:LenQry)
C 1080         Format(' <',I2,'> ',A,': ',\) ---backslash---
 1080        Format('$<',I2,'> ',A,': ')
            Call Value(Answer,IER)
            IF (IER.NE.0) Goto 10
         ENDIF
      Else If (Query(5:5).EQ.'C') THEN                             {Comment - just display line, no response possible}
         Call StrLen(Query(8:),Length)
         If (Query(6:6).EQ.'C') THEN                               {If comment is to be centered, left margin is 1/2 of
            Left = (80 - Length) / 2                               { nonused spaces}
         ELSE If (Query(6:6).EQ.'R') THEN                          {if it is to be right-justified, then margin is that
#}
            Left = (80-Length)
         ELSE                                                      {otherwise it starts at the left margin+1}
            Left = 1
         ENDIF
         Write(*,1090) Blank(1:Left),Query(8:)
 1090      Format(2A)
      ENDIF
      If (Query(6:6).EQ.'?') THEN
         Query(6:6) = 'D'
         Goto 10
      ENDIF
      Return
      END
      Logical FunctionRel2Lo(Val)
C-------------------------------------------------------------------------------------------------------------------------------
C-------------------------------------------------------------------------------------------------------------------------------
      If (NINT(Val).EQ.1) THEN
        Rel2Lo = .true.
      ELSE
        Rel2Lo = .false.
      ENDIF
      Return
      END
      SUBRoutine Shftxt(String)
C-------------------------------------------------------------------------------------------------------------------------------
C This routine shifts a dtring over to the left and pads the rest with blanks
C-------------------------------------------------------------------------------------------------------------------------------
C
      Character*(*) String
      Call StrBeg(String,Istart)
      Call StrLen(String,Length)
```

```
      If (Length+1-Istart.GT.0) THEN
        If (Istart.GT.1) THEN
          String(1:Length+1-Istart) = String(Istart:Length)
          Do 10 I = Length+2-Istart,Length
   10       String(I:I) = ' '
        ENDIF
      ENDIF
      Return
      END
      Logical Function Str2Lo(Text)
C-----------------------------------------------------------------------------------------------------------------------
C-----------------------------------------------------------------------------------------------------------------------
      Character*(*) Text
      Call StrCap(Text)
      Call StrBeg(Text,Istart)
      If (Istart.LE.0) THEN
        Str2Lo = .false.
      ELSE
        If (Text(Istart:Istart).EQ.'Y') THEN
          Str2Lo = .true.
        ELSE
          Str2Lo = .false.
        ENDIF
      ENDIF
      Return
      END
      SUBROUTINE STRBEG(STRING,LSTART)
C-----------------------------------------------------------------------------------------------------------------------
C         returns the first nonblank character position of STRING as LSTART
C-----------------------------------------------------------------------------------------------------------------------
C
      CHARACTER*(*) STRING
      LENGTH = LEN(STRING)
      Lstart = Length
      DO 10 I = 1,LENGTH
         IF (STRING(I:I).NE.' ') THEN
            LSTART = I
            Goto 11
         ENDIF
   10 CONTINUE
   11 RETURN
      END
      SUBRoutine StrCap(String)
C-----------------------------------------------------------------------------------------------------------------------
C         Capitialize the String
C-----------------------------------------------------------------------------------------------------------------------
C
      CHaracter*(*) String
      Call StrLen(String,Length)
      IF (Length.GT.0) THEN
        Do 10 I = 1,Length
   10     If (String(I:I).GE.'a'.AND.String(I:I).LE.'z') String(I:I) =
     &          Char(Ichar('A') + Ichar(String(I:I))-Ichar('a'))
      ENDIF
      Return
      END
C
      SUBROUTINE STRLEN(STRING,Length)
C-----------------------------------------------------------------------------------------------------------------------
C         returns as LENGTH the position of the last non-blank char. in STRING
C-----------------------------------------------------------------------------------------------------------------------
C
      CHARACTER*(*) STRING
      LENGTH = 0
      LstChr = Len(String)
      DO 10 I = 1,LstChr
         J = LstChr + 1 - I
         IF (STRING(J:J).NE. ' ') THEN
            Length = J
            Goto 11
         ENDIF
   10 CONTINUE
   11 RETURN
      END
      SUBROUTINE StrVal(Respon,ANSWER,IER)
C-----------------------------------------------------------------------------------------------------------------------
C         Converts the String Respon into a real value (0 if there is a problem)
C-----------------------------------------------------------------------------------------------------------------------
      CHARACTER*(*) RESPON,ValStr*21,Point*1,StrArg*21,SYM*1
      IER = 0
      Answer = 0
      Point = ' '
      Call StrBeg(Respon,Istart)
      Call StrLen(Respon,Length)
      If (Length.LE.0) Goto 30
      Do 10 I = Istart,Length
        If (Respon(I:I).EQ.'.') Goto 20
   10 Continue
      Point = '.'
   20 If (Length-Istart+1.GT.20) Length=Istart+19
      StrArg = Respon(Istart:Length)//Point
      Write(ValStr,1000) StrArg
 1000  Format(A)
      Read(ValStr,1010,Err=30) Answer
 1010  Format(G21.5)
      Goto 40
   30 IER = 1
   40 Return
      END
      SUBROUTINE VALUE(ANSWER,IER)
C-----------------------------------------------------------------------------------------------------------------------
```

```
C        reads a real value from the keyboard. IER = 0 if no error, 1 otherwise.
C-------------------------------------------------------------------------------------------------------------------
      CHARACTER RESPON*20
      Answer = 0
      READ(*,1000) RESPON
 1000   FORMAT(A)
      Call StrLen(Respon,Length)
      If (Length.GT.0) THEN
        Call StrVal(Respon,Answer,IER)
      ELSE
        IER = 1
      ENDIF
      Return
      END
      SUBROUTINE VTAB(LINERQ)
C-------------------------------------------------------------------------------------------------------------------
C        moves the cursor to the leftmost position on CRT line LINERQ
C-------------------------------------------------------------------------------------------------------------------
C
      Write(*,1000)CHAR(27)//'['//CHAR(ICHAR('0')+INT((LINERQ-1)/10))
     &             //CHAR(ICHAR('0')+Mod((LineRq-1),10))//';0H'
 1000 FORMAT(1X,A)
      RETURN
      END
      SUBRoutine Wait4R
C-------------------------------------------------------------------------------------------------------------------
C        displays a prompt in line 22 & waits for a return key press
C-------------------------------------------------------------------------------------------------------------------
C
      Character*1 respon
      Call Notice('\Press [RETURN] to continue: ',.false.,23)
      Read(*,1010) Respon
 1010  Format(A)
      Return
      END
      SUBROUTINE WARNIN(Messag,Clear,Line)
C-------------------------------------------------------------------------------------------------------------------
C This subroutine displays the MESSAge centered on LINE and will clear the screen if CLEAR  is true. It waits for a Response
C before before returning control.
C-------------------------------------------------------------------------------------------------------------------
C
      Character*(*) Messag
      Character*80 Blank
      Logical Clear
      Data Blank/' '/
      Call Strlen(Messag,Length)
      Margin = (80 - Length) / 2
      If (Margin.LE.0) Margin = 1
      If (Clear) Call Page('Dummy')
   10 Call Vtab(Line)
        Write(*,1000)Blank(1:Margin),Messag(1:Length)
 1000     Format(2A)
      Call Wait4R
      Return
      END
      Character*3 Function YesNo(Boolen)
C-------------------------------------------------------------------------------------------------------------------
C-------------------------------------------------------------------------------------------------------------------
      Logical Boolen
      If (Boolen) THEN
        YesNo = 'Yes'
      ELSE
        YesNo = ' No'
      ENDIF
      Return
      END
```